

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

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE

FACULTE D'ELECTRONIQUE ET D'INFORMATIQUE



MEMOIRE

Présenté pour obtenir le diplôme de : MAGISTER

En : INFORMATIQUE

Spécialité : INFORMATIQUE MOBILE

Par : M^{elle} Amel BERRACHEDI

Thème :

Modélisation de la notion d'énergie dans les réseaux de capteurs à l'aide des réseaux de Petri

Soutenu publiquement le 28 juin 2012 devant le jury composé de :

M ^{me} Samira MOUSSAOUI	MCA à l'USTHB	Présidente
M ^{me} Malika BOUKALA-IOUALALEN	Professeur à l'USTHB	Directrice de mémoire
M ^{elle} Nabila SALMI	MCA à l'USTHB	Examinatrice
M ^r Abdelouahid DERHAB	MRA au CERIST	Examineur

Dédicaces

À mon père, à ma mère,

À mon frère, mes soeurs, mes beaux frères et mes neveux,

À notre illustre Imam Cheikh M^{ed} Seghir BENBETKA,

À toute ma famille et mes amis,

À tous ceux qui me sont chers,

Je dédie ce travail.

Remerciements

Je tiens à exprimer mes vifs remerciements envers toutes les personnes qui ont contribué au bon déroulement de ce travail.

Mes premiers remerciements et ma grande gratitude s'expriment envers Professeur M. BOUKALA-IOUALALEN, ma directrice de thèse qui a fait preuve de disponibilité en toute circonstance. Ses conseils, ses encouragements et ses critiques ont été d'une précieuse aide pour l'accomplissement de ce projet.

Je tiens à remercier très chaleureusement Docteur S. MOUSSAOUI, Docteur N. SALMI et Docteur A. DERHAB d'avoir accepté de juger ce travail.

Je remercie vivement tous mes collègues de l'équipe MEPQOS pour leur sympathie durant le temps où on a travaillé ensemble et pour leurs précieuses indications et discussions de recherche dans le domaine des réseaux de Petri et des réseaux de capteurs.

Je pourrai écrire un cinquième chapitre rien que pour remercier mes parents qui m'ont apporté le courage et la motivation nécessaires pour aller jusqu'au bout. Grand merci, de m'avoir soutenue et conseillée au cours de toutes ces années d'études tant sur le plan technique que humain.

Un remerciement particulier à ceux qui ont été continuellement présents et m'ont épaulée, de près ou de loin, par leur aide à la réalisation de ce travail.

Résumé

Les progrès récents dans les communications sans fil et le domaine de l'électronique ont permis le développement des micro-capteurs, moins coûteux et multifonctionnels. Ces caractéristiques ont permis de se projeter dans la naissance des réseaux de capteurs sans fil, et de favoriser leur utilisation dans une multitude d'applications. Cette large gamme d'applications accroît la complexité de tels réseaux et engendre encore beaucoup de défis liés à la limitation des capacités de traitement, de stockage et surtout d'énergie. De telles contraintes nécessitent une phase préliminaire de vérification afin de garantir le niveau de fiabilité requis par les applications.

Notre projet s'inscrit dans la thématique de la modélisation et la vérification formelle dans les réseaux de capteurs. Nous montrons dans un premier lieu les caractéristiques essentielles des modèles formels. Nous étudions ensuite un certain nombre de formalismes tels que les réseaux de neurones, les chaînes de Markov, les algèbres de processus, les automates temporisés ainsi que les réseaux de Petri. A l'issue de cette étude, notre choix s'est porté sur ce dernier modèle qui semble le plus adéquat pour notre travail.

Notre contribution porte sur la quantification de la consommation d'énergie sur le modèle réseaux de Petri. Pour atteindre cet objectif, nous proposons le formalisme EgPN (*Energy Petri Net*) qui intègre et modélise le facteur énergie de manière explicite.

Mots clés : Réseaux de capteurs, réseaux de Petri, conservation d'énergie, routage.

Table des matières

Introduction générale	1
1 Réseaux de Capteurs Sans Fil : Concepts, Routage, Défis	5
1.1 Introduction	5
1.2 Définitions et concepts fondamentaux	6
1.2.1 Qu'est ce qu'un noeud capteur ?	6
1.2.2 Qu'est ce qu'un réseau de capteurs ?	8
1.3 Applications et classification	9
1.4 Architecture protocolaire	10
1.5 Facteurs de conception des protocoles de routage	11
1.5.1 Limitations des ressources énergétiques et physiques	12
1.5.2 Topologie du réseau	12
1.5.3 Densité du déploiement	12
1.5.4 Scalabilité et tolérance aux pannes	13
1.5.5 Sécurité	13
1.5.6 Modèle de liens de communication	14
1.5.7 Coût de production	14
1.6 Métriques de routage et durée de vie du réseau	15
1.7 Conclusion	16
2 Réseaux de Capteurs Sans Fil : Routage, Conservation d'énergie, Clustering	17
2.1 Introduction	17
2.2 Techniques de conservation d'énergie	17
2.2.1 Les communications multi-sauts	18
2.2.2 L'agrégation de données	19
2.2.3 Le clustering	20
2.3 Protocoles de routage hiérarchiques	22
2.4 Protocole de routage LEACH	24
2.4.1 Fonctionnement du protocole LEACH	24
2.4.1.1 Phase d'initialisation	25

2.4.1.2	Phase de transmission	25
2.4.2	Modèle radio du protocole LEACH	26
2.5	Nécessité de la vérification formelle	28
2.5.1	Pourquoi évaluer les performances d'un système?	28
2.5.2	De la simulation vers les modèles formels	28
2.6	Conclusion	29
3	Réseaux de Capteurs Sans Fil : Modélisation Formelle	31
3.1	Introduction	31
3.2	Caractéristiques d'un modèle formel	31
3.3	Modèles formels pour les réseaux de capteurs	32
3.3.1	Modèles à base de réseaux de neurones	33
3.3.1.1	Présentation des réseaux de neurones	33
3.3.1.2	Réseaux de neurones pour les réseaux de capteurs	35
3.3.2	Modèles à base de chaînes de Markov	39
3.3.2.1	Présentation des chaînes de Markov	40
3.3.2.2	Chaînes de Markov pour les réseaux de capteurs	40
3.3.3	Modèles à base d'algèbres de processus	43
3.3.3.1	Présentation des algèbres de processus	43
3.3.3.2	Algèbres de processus pour les réseaux de capteurs	44
3.3.4	Modèles à base d'automates temporisés	48
3.3.4.1	Présentation des automates temporisés	48
3.3.4.2	Automates temporisés pour les réseaux de capteurs	50
3.3.5	Modèles à base de réseaux de Petri	53
3.3.5.1	Présentation des réseaux de Petri	53
3.3.5.2	Réseaux de Petri pour les réseaux de capteurs	55
3.4	Discussion et problématique	59
3.5	Conclusion	60
4	Modèle formel pour la consommation d'énergie : Energy Petri Net	62
4.1	Introduction	62
4.2	Formalisation du modèle	63
4.3	Etude comportementale - Aspects dynamiques	65
4.3.1	Tir de transitions	65
4.3.2	Graphe des marquages	66
4.3.3	Graphe d'états	68
4.4	Etude structurelle - Algèbre linéaire	69
4.4.1	Représentation des EgPN par matrices	70
4.4.2	Représentation des tirs de transitions	71

4.5	Exemple de modélisation	73
4.5.1	Modèle EgPN pour le clustering statique	75
4.5.2	Modèle EgPN pour le clustering dynamique	82
4.6	Conclusion	88
	Conclusion générale	90
	Perspectives et travaux futurs	93

Table des figures

1.1	Architecture d'un noeud capteur [100].	6
1.2	Consommation d'énergie par les unités d'un noeud capteur [12].	7
1.3	Architecture de communication d'un réseau de capteurs [100].	8
1.4	Pile protocolaire des réseaux de capteurs [40].	10
2.1	Transmissions directes.	18
2.2	Communications multi-sauts [18].	19
2.3	Transmission sans agrégation de données [18].	19
2.4	Transmission avec agrégation de données [18].	20
2.5	Architectures de communication adoptées par les réseaux de capteurs [40].	22
2.6	Diagramme de la phase d'initialisation du protocole LEACH [12].	26
2.7	Modèle de liens de communication [20].	27
3.1	Analogie neurone biologique et neurone formel.	33
3.2	Un neurone formel.	34
3.3	Exemple d'un réseau de neurones formels.	34
3.4	Noeud capteur vs neurone formel. [65]	36
3.5	Classification de données par un réseau de neurones pour la détection des incendies [101].	38
3.6	Exemple d'une chaîne de Markov [85].	40
3.7	Evaluer l'énergie consommée par la CPU d'un noeud capteur [85, 86].	42
3.8	Portée de communication entre les processus [67].	46
3.9	Système Horloge/Utilisateur modélisé par un automate temporisé [95].	49
3.10	Exemple d'un automate temporisé à deux horloges [95].	50
3.11	Automate temporisé modélisant la détection d'un noeud radio-actif [104].	51
3.12	Automate temporisé modélisant la technique d'inondation [104].	52
3.13	Exemple d'un RdP.	53
3.14	Modélisation du comportement d'un noeud adoptant l'algorithme de LEACH à l'aide d'un RdPS [39].	56
3.15	Modélisation d'un algorithme d'élection des CH à l'aide d'un RdPC [53].	57

4.1	Exemple d'un EgPN.	64
4.2	Envoi et réception des données.	66
4.3	Graphe des marquages associé à l'exemple de la figure 4.2.	67
4.4	Graphe d'états associé à l'exemple de la figure 4.2.	68
4.5	Modèle d'une communication entre deux noeuds capteurs.	71
4.6	Modèle EgPN pour le clustering statique.	75
4.7	Modèle EgPN d'un noeud CH selon le clustering statique.	80
4.8	Modèle EgPN d'un noeud non-CH selon le clustering statique.	80
4.9	Modèle EgPN pour le protocole LEACH.	83

Liste des tableaux

4.1	Paramètres du réseau	74
4.2	Définitions des places du modèle du clustering statique.	76
4.3	Définitions des transitions du modèle du clustering statique.	76
4.4	Energies seuils des transitions du modèle du clustering statique.	76
4.5	Définitions des places du modèle de LEACH.	84
4.6	Définitions des transitions du modèle de LEACH.	84
4.7	Energies seuils des transitions du modèle de LEACH.	84
4.8	Consommation d'énergie dans les deux modes de clustering.	88

Introduction générale

Depuis leur apparition, les réseaux sans fil envahissent de plus en plus le monde informatique. Les ondes radioélectriques, qui ont pour but de régler les problèmes de câblage, servent encore à l'ubiquité de l'information et à la facilité de déploiement.

On a d'abord assisté à la croissance exponentielle des réseaux cellulaires qui dépendent largement des infrastructures fixes. Le problème qui se pose est que les points d'accès peuvent être soumis à des risques de destruction dans certaines applications (comme le domaine militaire). Par conséquent, la nature très versatile de ces réseaux rend leurs performances relativement faibles et mal adaptées aux applications industrielles. Pour avoir une plus grande souplesse et robustesse, un autre type d'architecture de réseaux intitulé ad hoc a donc été conçu. Leur intérêt majeur est qu'ils ne requièrent pas une administration centralisée et n'exigent pas une infrastructure préexistante. Toutefois, les réseaux sans fil actuels ont besoin d'une plus grande capacité de collecter et de traiter l'information, c'est-à-dire, de fournir une certaine intelligence au sein du réseau [84]. C'est pour cela que les réseaux ad hoc trouvent actuellement une application dans les réseaux de capteurs sans fil ou WSNs (*Wireless Sensor Networks*); l'une des évolutions les plus importantes qui attirent l'attention de l'industrie ainsi que la communauté des chercheurs. Ce sont des réseaux utilisés dans une large gamme d'applications comme la surveillance de l'environnement, l'exploration robotique, la gestion du trafic, les applications militaires, les systèmes médicaux, les opérations de secours et ainsi de suite [35].

Les réseaux de capteurs sans fil sont constitués d'un noeud collecteur appelé station de base ou noeud puits qui est chargé de récupérer et rassembler les données collectées et acheminées par un ensemble de noeuds capteurs miniatures, autonomes, multifonctionnels et distribués sur une zone de captage afin de mesurer une grandeur physique ou surveiller un évènement [69, 63]. Chaque noeud capteur est composé de trois unités alimentées par une batterie embarquée et très faible [12]. La première unité est responsable du captage des données qui seront par la suite traitées par l'unité de calcul (ou de traitement). La troisième unité, appelée unité de transmission, est responsable de toutes les émissions/réceptions entre les noeuds capteurs.

Des mesures expérimentales ont montré que c'est la transmission des données qui est la plus consommatrice en énergie, et de façon significative, alors que le captage et les calculs consomment très peu [65]. Comme les noeuds capteurs sont alimentés principalement par des batteries à capacités très réduites et généralement irremplaçables, ils doivent alors fonctionner avec un bilan énergétique économe. L'objectif principal des réseaux de capteurs est donc de trouver des moyens pour une gestion efficace en termes d'énergie et pour une transmission fiable des données de la source vers la destination de sorte que la durée de vie du réseau soit maximisée, c'est ce que l'on appelle problème de routage [100]. Cette problématique conjuguée à l'objectif de maximisation de la durée de vie du réseau a suscité de nombreux travaux de recherche.

Afin de concevoir des protocoles de routage efficaces en énergie, il est extrêmement important de faire d'abord une analyse des différents facteurs provoquant la dissipation de l'énergie d'un noeud capteur. Pour y remédier, des techniques de conservation d'énergie ont été proposées comme la communication multi-sauts qui permet de réduire les distances de transmission, l'agrégation de données qui permet de réduire le flux de données propagées, et enfin, la technique du clustering qui englobe les deux premières approches [40].

La technique du clustering consiste à hiérarchiser le réseau et l'organiser en groupes ou *clusters*. Chacun est administré par un seul noeud appelé *Cluster Head* ou CH. Seuls les CH communiquent avec la station de base (qui est supposée éloignée) et effectuent les tâches les plus coûteuses, énergiquement parlant, comme la gestion des clusters et l'agrégation de données provenant des autres noeuds non-CH qui se consacrent uniquement au captage. Parmi les protocoles de routage adoptant cette technique, on cite le protocole LEACH (*Low Energy Adaptive Clustering Hierarchy*) [20] qui est le premier protocole de routage hiérarchique conçu pour les réseaux de capteurs.

Avant de mettre en oeuvre ces protocoles dans le monde réel, une phase d'analyse et d'évaluation s'avère nécessaire afin de prouver leur efficacité et qu'ils possèdent bien les propriétés voulues par leur concepteur [36]. Ce qui permet de réduire le nombre de défaillances possibles et d'augmenter la confiance que l'on peut accorder à ces protocoles. L'approche classique consiste à évaluer les protocoles de routage par des simulateurs. Toutefois, la nature des applications des réseaux de capteurs est très importante pour l'évaluation des performances des protocoles réseau. En effet, certaines applications comme la détection de la température d'un champ agricole acquièrent moins d'importance que d'autres comme les applications militaires et les centrales nucléaires. Ces dernières applications sont dites critiques.

D'une manière générale, un protocole peut être considéré comme correct en utilisant les simulations, mais, de nombreux incidents ont prouvé que cette notion de correction n'est pas suffisante [25] : le bogue du Pentium en 1994 (480 millions de dollars) et l'explosion du premier vol d'Ariane5 en 1996 (500 millions de dollars) étant parmi les exemples les plus coûteux. En plus des dépenses, de telles erreurs peuvent entraîner des conséquences plus graves comme le problème de l'appareil d'irradiation thérapeutique Therac-25 (2 morts et 4 irradiés). Dans le cadre des réseaux de capteurs, la durée de vie du réseau occupe une place exceptionnelle car ces réseaux doivent le plus souvent avoir une durée de vie de l'ordre de plusieurs mois, voire de quelques années, puisque le remplacement des batteries n'est pas une option envisageable pour des réseaux avec des milliers de noeuds déployés dans des environnements hostiles et éloignés. Ainsi, une simple faille qui risque de survenir dans un protocole de routage peut réduire considérablement la durée de vie du réseau et peut avoir des répercussions néfastes sur le coût de déploiement du réseau. Pour cela, l'utilisation de techniques formelles devient incontournable.

Le principal objectif de notre projet consiste à présenter une étude sur les modèles formels qui existent dans la littérature et qui sont utilisés pour la description des réseaux de capteurs. Différents formalismes ont été récemment utilisés dans la modélisation et l'analyse de plusieurs concepts dans les réseaux de capteurs afin de garantir le niveau de fiabilité et de sûreté de fonctionnement requis par les applications. Parmi ces formalismes, nous citons les chaînes de Markov (1907) [33], les réseaux de neurones formels (1943) [76], les réseaux de Petri (1962) [75], les algèbres de processus (1973) [24], les automates temporisés (1990) [4]...

Etant plus anciens que les réseaux de capteurs apparus en 1999 [12], ces modèles formels ne prennent pas la consommation d'énergie comme étant une préoccupation majeure. Ces modèles permettent cependant de modéliser d'autres aspects comme la notion du temps, alors que, la quantification de l'énergie n'apparaît pas. En effet, les réseaux de neurones formels sont beaucoup plus utilisés pour améliorer les QoS dans un réseau de capteurs et améliorer la fiabilité de transmission. En revanche, les algèbres de processus sont plus expressives ; elles ne se restreignent pas à la QoS mais elles touchent à beaucoup de notions des réseaux de capteurs comme la consommation d'énergie durant le routage. Mais reste à dire que ce sont des techniques très difficiles à manipuler parce qu'elles assurent la caractéristique d'expressivité au détriment de la simplicité. Les chaînes de Markov quant à elles sont plus appropriées pour l'évaluation de performances (comme la durée de vie du réseau) des protocoles de routage. Elles ne permettent cependant pas la description des interactions et des communications qui peuvent exister entre

les noeuds capteurs. Les automates temporisés peuvent à leur tour, estimer la durée de vie d'un réseau de capteurs, mais, la quantification de l'énergie n'apparaît pas sur le modèle. Concernant les réseaux de Petri, certains travaux ont évalué la consommation d'énergie à l'aide des outils stochastiques, mais, cette métrique a été modélisée de manière implicite.

Ce mémoire tente d'apporter une contribution en s'intéressant particulièrement au mécanisme du routage dans les réseaux de capteurs soumis à des contraintes énergétiques. Cette contribution consiste en premier lieu à enrichir l'un des modèles formels étudiés avec la dimension énergétique. En deuxième lieu, nous nous intéressons à analyser l'impact de la couche routage sur la durée de vie d'un réseau de capteurs en évaluant la consommation d'énergie à l'aide du modèle résultant. Notre choix s'est porté sur les modèles à base de réseaux de Petri en raison des multiples avantages qu'ils présentent et parce qu'ils pourraient être les plus appropriés pour les spécificités de notre problématique. Le modèle résultant intitulé EgPN (*Energy Petri Net*) est un type de réseau de Petri où la consommation d'énergie est modélisée d'une manière explicite.

Le manuscrit s'articule, après une introduction, autour de quatre chapitres dont les deux premiers sont consacrés aux rappels et aux notations essentiels pour la suite ; on y trouve une description détaillée des réseaux de capteurs sans fil, un état de l'art sur les différentes techniques de minimisation de la consommation d'énergie, ainsi que les protocoles de routage hiérarchiques. La méthodologie adoptée et le schéma général de la contribution font l'objet du chapitre trois, différentes méthodes formelles y sont présentées et étudiées. Notre apport est consigné au chapitre quatre, on y a introduit et étudié le modèle EgPN. Ce modèle permet de modéliser explicitement l'énergie consommée dans le réseau ; on y présente deux techniques du clustering (le clustering statique et le protocole de routage LEACH). Etant donné que les noeuds capteurs ont le même comportement, nous avons eu recours à la notion de symétrie afin de réduire les modèles EgPN des deux techniques du clustering. Le manuscrit se termine par une conclusion générale faisant ressortir certaines perspectives de recherche. Ce travail a nécessité une recherche bibliographique très dense, une bonne partie a été mentionnée.

Chapitre 1

Réseaux de Capteurs Sans Fil : Concepts, Routage, Défis

1.1 Introduction

Les progrès récents dans les domaines de l'électronique et de la communication ont permis à des usagers munis d'unités sans fil d'accéder à l'information indépendamment du temps et de leurs positions géographiques [43]. En plus de cette notion d'ubiquité, la miniaturisation des dispositifs électroniques et la diminution des coûts de production ont permis de se projeter dans la naissance des réseaux de capteurs sans fil qui sont des réseaux apparentés aux réseaux ad hoc du point de vue infrastructure, architecture, autonomie d'énergie et utilisation des liens sans fil pour communiquer.

Ces développements technologiques favorisent l'utilisation des réseaux de capteurs dans une multitude d'applications. Celles-ci nécessitent souvent un déploiement dense et aléatoire dans des environnements hostiles et éloignés où les noeuds capteurs deviennent inaccessibles et doivent pouvoir survivre dans les conditions les plus extrêmes dictées par leur environnement d'utilisation. Il est donc nécessaire d'avoir des stratégies de communication efficaces qui essaient de réduire au maximum la perte d'énergie des noeuds capteurs afin d'augmenter la durée de vie du réseau. On appelle ces stratégies, protocoles de routage qui sont des algorithmes permettant le bon acheminement des paquets à travers le réseau avec une latence faible, peu de charge du réseau (minimiser les paquets transmis), et bien évidemment, peu d'énergie. Cet objectif est compliqué d'avantage à cause de l'absence d'infrastructure de communication fixe, en plus de limitations matérielles imposées par la taille miniaturisée des capteurs.

Avant d'entrer dans l'étude du routage et de la conservation d'énergie dans les réseaux de capteurs, nous présentons dans ce chapitre ce que sont les réseaux de capteurs, leurs applications et leurs classifications. Ensuite, nous expliquons les différents défis et facteurs de conception des protocoles de routage dans un réseau de capteurs, notamment la durée de vie du réseau.

1.2 Définitions et concepts fondamentaux

Avant de donner la définition des réseaux de capteurs, il s'avère plus judicieux de donner certains concepts sur les noeuds capteurs.

1.2.1 Qu'est ce qu'un noeud capteur ?

Un noeud capteur est un mini-composant électronique, qui permet d'acquérir des événements réels sur son environnement (température, mouvement, etc.), les traiter et les communiquer à un centre de contrôle. [40]

Un noeud capteur est composé de trois unités alimentées par une source d'énergie autonome et limitée. Ces unités sont illustrées dans la figure 1.1.

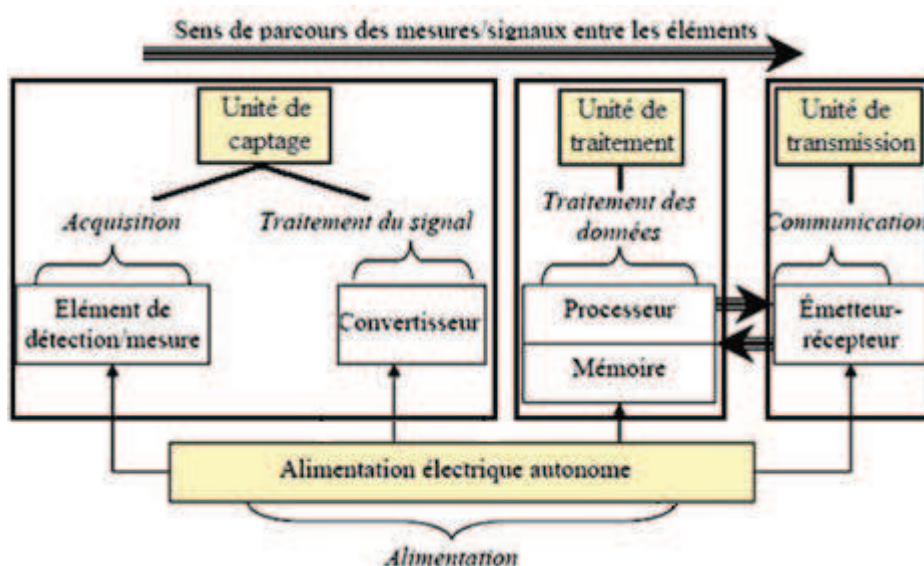


FIGURE 1.1 – Architecture d'un noeud capteur [100].

- **Unité de captage** : Elle est composée de deux sous-unités : une unité d'acquisition, qui permet de détecter les mesures désirées par un capteur, et d'une unité de traitement des signaux qui transforme les mesures analogiques détectées en signaux

numériques par un convertisseur analogique numérique.

- **Unité de traitement** : Elle est composée de deux sous-unités : unité de stockage (une mémoire) et d'une unité de calcul (un processeur) qui permet d'effectuer des calculs simples pour que ce noeud puisse collaborer avec les autres noeuds du réseau. De plus, elle possède deux interfaces :
 - La première, liée à l'unité de captage par laquelle, elle reçoit les mesures détectées.
 - La seconde, liée à l'unité de transmission par laquelle, elle communique les données qu'elle a traitées.

- **Unité de transmission (*Transceiver*)** : Elle est responsable de toutes les émissions et réceptions de données via un support de communication sans fil qui relie le noeud au réseau. L'unité de transmission est l'unité qui consomme le plus d'énergie par rapport aux autres unités (voir figure 1.2). Dans la littérature, il a été prouvé que le coût de transmission de données énergiquement parlant, peut atteindre un taux de 70% par rapport au coût général du fonctionnement d'un noeud capteur. D'autres ouvrages ont démontré que l'énergie dissipée lors d'une transmission d'un seul paquet de données équivaut à 1000 opérations arithmétiques [65].

Outre ces unités, il existe des capteurs qui sont dotés d'autres composants additionnels suivant les domaines d'applications. On cite à titre d'exemple, les systèmes de localisation GPS (*Global Positioning System*) ou les mobilisateurs chargés de déplacer les noeuds capteurs en cas de nécessité.

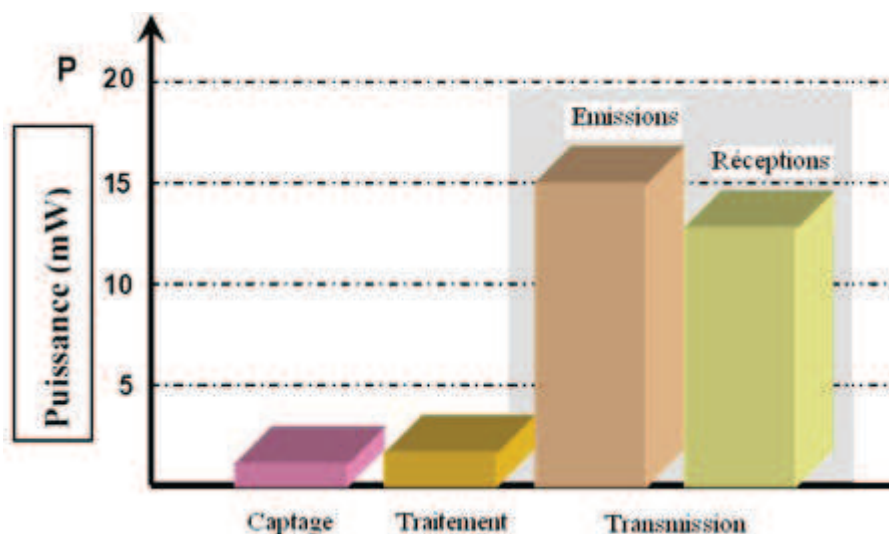


FIGURE 1.2 – Consommation d'énergie par les unités d'un noeud capteur [12].

1.2.2 Qu'est ce qu'un réseau de capteurs ?

Les noeuds capteurs qui coopèrent entre eux forment une infrastructure de communication appelée réseau de capteurs. Ce dernier se compose généralement d'un nombre très élevé de noeuds capteurs, qui communiquent entre eux via des liens sans fil de type radiofréquence ou de type optique. Ces noeuds ne sont pas intégrés à une quelconque architecture de communication préexistante, mais ils communiquent à l'aide d'un réseau ad hoc sans fil. Leur déploiement peut être déterministe (ils sont posés à des endroits précis à la main ou par robots), comme il peut être aléatoire (souvent dispersés par voie aérienne à l'aide d'avions ou hélicoptères) [40].

Après leur déploiement, généralement aléatoire, sur une certaine zone de captage, les noeuds capteurs s'auto-organisent afin de construire l'architecture de communication appropriée qui est intimement liée aux domaines d'applications (elle prend généralement la forme illustrée sur la figure 1.3). Selon une communication multi-sauts, les noeuds capteurs sont chargés de collecter des données et les acheminer vers un noeud collecteur appelé station de base ou noeud puits. Ce dernier analyse ces données et transmet à son tour l'information finale à l'utilisateur via internet ou bien satellite.

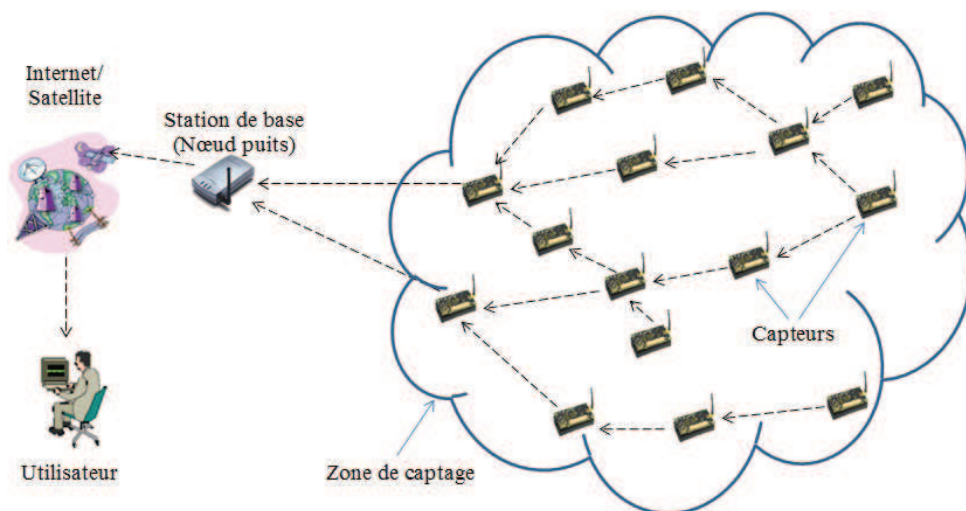


FIGURE 1.3 – Architecture de communication d'un réseau de capteurs [100].

1.3 Applications et classification

Comme beaucoup de développements technologiques, les réseaux de capteurs sans fil ont émergé pour des besoins militaires. Puis, ils ont trouvé leur chemin pour des applications civiles. De nos jours, les réseaux de capteurs sans fil se révèlent très utiles dans plusieurs domaines d'applications. Ils peuvent par exemple aider à avoir un bon système de sécurité à la maison, réaliser des opérations de sauvetage en localisant les survivants, détecter des incendies dans une forêt, surveiller la température et l'humidité d'un champ agricole, surveiller la présence des médecins dans un hôpital, etc.

En raison des domaines d'applications des réseaux de capteurs, les noeuds capteurs sont souvent stationnaires ce qui rend le trafic dans ces réseaux statique (en les comparant aux réseaux ad hoc) [11]. Ces applications qui constituent une différence clé entre ces deux types de réseaux [69] permettent de classer les réseaux de capteurs en trois catégories [104] :

- La première classe contient les réseaux de collection des données d'environnement qui ont pour but la collection périodique des données telles que la température, l'humidité d'un champ agricole, puis leurs transmissions au noeud puits. Ce type de réseau ne présente pas de contraintes de type temps de latence et nécessite généralement des noeuds capteurs statiques.
- La deuxième classe englobe les réseaux de surveillance et de sécurité. Elle permet de contrôler continuellement la détection d'une anomalie dans l'environnement comme la détection d'un incendie dans une forêt. Ainsi, les noeuds capteurs transmettent seulement les rapports concernant une violation de la sécurité durant un excellent temps de latence. Bien qu'elles soient totalement différentes sur le plan applicatif, ces deux premières catégories de réseaux ont un point commun : les noeuds capteurs ne sont pas forcément mobiles.
- Les réseaux de poursuite constituent la troisième classe d'applications. Ils sont généralement utilisés dans le domaine militaire. Ce type de réseau est caractérisé par une topologie mobile qui change d'une manière fréquente et imprévisible suite à la mobilité continue des noeuds capteurs. Ainsi, la mobilité dans les réseaux de capteurs découle des applications où les noeuds sont en perpétuel mouvement comme dans les réseaux sous-marins, les réseaux militaires et les réseaux véhiculaires [5, 50].

1.4 Architecture protocolaire

Comme tout autre réseau de télécommunication, les réseaux de capteurs adhèrent à une architecture en couches afin d'améliorer sa robustesse. Une pile protocolaire de cinq couches est adoptée par les noeuds du réseau. Citons la couche application, la couche transport, la couche réseau, la couche liaison de données et la couche physique (figure 1.4).

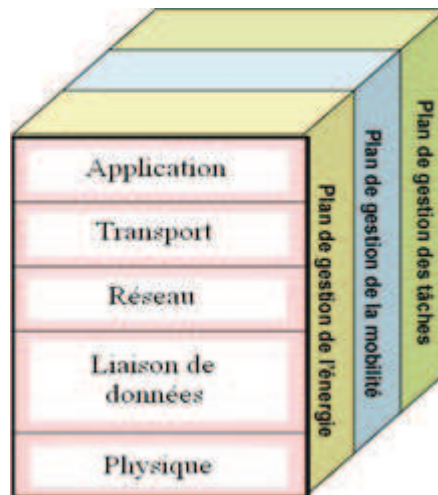


FIGURE 1.4 – Pile protocolaire des réseaux de capteurs [40].

1. **Couche application** : Elle assure l'interface avec les applications. Il s'agit donc de la couche la plus proche des utilisateurs, gérée directement par les logiciels.
2. **Couche transport** : Elle vérifie le bon acheminement des données et la qualité de la transmission. Dans les réseaux de capteurs, la fiabilité de transmission n'est pas majeure. Ainsi, les erreurs et les pertes sont tolérées.
3. **Couche réseau** : Elle s'occupe de l'acheminement (ou **routage**) de données fournies par la couche transport. Elle établit les routes entre les noeuds capteurs et le noeud puits et sélectionne le meilleur chemin en termes d'énergie, délai de transmission, débit, etc.
4. **Couche liaison de données** : Elle est constituée de deux sous-couches :
 - la couche LLC (*Logical Link Control*) qui fournit le contrôle de flux et la détection et la correction d'erreurs intervenues sur la couche physique.
 - la couche MAC (*Media Access Control*) qui est responsable de la synchronisation des trames et de l'accès au media physique afin d'éviter le problème de collision de données.
5. **Couche physique** : Elle permet de moduler les données et les acheminer dans le media physique tout en choisissant les bonnes fréquences.

En plus de ces cinq couches, la pile protocolaire possède trois plans (niveaux) de gestion : le plan de gestion des tâches qui répartit les tâches entre les noeuds capteurs, le plan de gestion de mobilité qui garde une image sur la localisation des noeuds, et, le plan de gestion de l'énergie qui aide à conserver le maximum d'énergie.

A cause de leur taille réduite, les batteries dont disposent les noeuds capteurs sont réduites et limitées. De plus, elles sont généralement irremplaçables étant donné que dans la plupart des applications, les noeuds capteurs sont placés aléatoirement et dans des environnements hostiles et distants. Il est donc très important que les batteries durent aussi longtemps que possible. Pour cela, l'énergie est la contrainte la plus précieuse puisqu'elle influe directement sur la durée de vie des noeuds capteurs, et donc, sur la durée de vie du réseau en entier. C'est pourquoi, le plan de gestion de l'énergie occupe une place importante sur toutes les couches de la pile protocolaire, particulièrement, sur la couche réseau vu que le routage et la transmission de données sont les tâches les plus coûteuses en termes d'énergie. L'objectif principal pour cette couche est de trouver des moyens pour une consommation d'énergie efficace et pour une transmission des données fiable à partir de la source vers la destination de sorte que la durée de vie du réseau soit maximisée. Maximiser la durée de vie du réseau consiste à minimiser la consommation énergétique des noeuds d'une part, et à trouver des moyens pour réduire les distances de transmissions et le nombre de messages échangés d'autre part [100].

Dans ce document, nous nous focalisons sur le routage de données. De nombreux algorithmes de routage, qui sont de plus en plus perfectionnés, ont été spécifiquement conçus pour les réseaux de capteurs où la consommation d'énergie est un facteur essentiel.

1.5 Facteurs de conception des protocoles de routage

Les réseaux de capteurs sans fil sont considérés comme étant une sous classe des réseaux mobiles ad hoc. Cependant, ils sont très différents les uns des autres et ils ont peu de propriétés en commun comme la communication sans fil et l'absence d'infrastructure fixe [69]. Ces dissemblances que l'on peut résumer dans cette section, rendent les solutions du routage ad hoc inadaptées aux réseaux de capteurs et influencent extrêmement le choix des protocoles de routage à implémenter.

1.5.1 Limitations des ressources énergétiques et physiques

Le premier point de différence entre les réseaux de capteurs et les réseaux ad hoc est que les noeuds capteurs sont fortement contraints en termes d'énergie et de capacité de calcul et de stockage.

En ce qui concerne la consommation d'énergie, elle est d'une très haute importance. D'une part, l'alimentation des noeuds capteurs est assurée par des batteries généralement irremplaçables à cause de l'environnement hostile où ils sont déployés [35]. D'autre part, l'émission et la réception des données, la réorganisation du réseau ainsi que le changement de sa topologie sont les tâches les plus coûteuses en termes d'énergie, ce qui rend sa gestion et sa conservation primordiale [11]. Dans ce cas, les protocoles de routage doivent être économes, performants et optimisés afin de maximiser la durée de vie du réseau [36]. Par ailleurs, la miniaturisation des capteurs rend les performances du réseau très limitées. Ces limitations imposent donc des portées de transmission réduites contraignant les informations à être relayées de noeud en noeud avant d'atteindre le destinataire. C'est la raison pour laquelle les réseaux de capteurs adoptent des communications multi-sauts [28].

1.5.2 Topologie du réseau

La topologie réseau peut être abordée de différentes manières. Il peut s'agir d'une topologie réseau physique qui peut être instable suite aux imprévus comme la disparition ou l'apparition des noeuds dans le réseau, ou bien, à cause de la mobilité des noeuds capteurs qui peuvent être attachés à des objets mobiles qui se déplacent d'une façon libre et arbitraire [5].

Comme il peut s'agir d'une topologie réseau logique auquel cas, l'architecture de communication capture le partitionnement du réseau en différentes régions. Chacune des régions possède des propriétés et des rôles distincts. Elle signifie la façon dont les noeuds sont regroupés et comment leurs informations seront acheminées [3]. La topologie réseau logique qui en résulte peut donc être dynamique et peut évoluer au cours du temps selon les hypothèses considérées par les protocoles de routage.

1.5.3 Densité du déploiement

La densité du déploiement est plus élevée dans les réseaux de capteurs que dans les réseaux ad hoc. Le nombre de noeuds capteurs peut atteindre les millions dans une petite surface pour permettre une meilleure granularité de surveillance. De plus, si plusieurs noeuds capteurs se trouvent dans une région, un noeud défaillant pourra être remplacé par un autre. Cependant, la densité de déploiement donne naissance à des challenges pour

la communication entre les noeuds. En effet, elle provoque des collisions ou des endommagements des paquets transmis. Afin de pallier à ce type de désagrément et pour garantir un bon fonctionnement du réseau, les protocoles de routage doivent être capables de gérer ce grand nombre de noeuds.

1.5.4 Scalabilité et tolérance aux pannes

A cause de l'épuisement des batteries, d'un dégât matériel ou d'une interférence environnementale, les noeuds capteurs sont assez fragiles et vulnérables à diverses formes de défaillances. Cependant, cette faible protection physique ne doit pas affecter le fonctionnement global du système. Ainsi, les protocoles de routage doivent être capables de maintenir les fonctionnalités du réseau et considérer la panne physique d'un noeud comme une propriété normale ; c'est ce qu'on appelle la tolérance aux pannes.

En cas d'ajout des noeuds, il faut également maintenir les fonctionnalités du système et ne pas mener à une dégradation de performances du réseau ; c'est ce qu'on appelle la scalabilité ou le passage à l'échelle.

Pour récapituler, les protocoles réseau ne devraient pas souffrir d'une dégradation de performances dans le cas d'endommagement de noeuds où dans le cas d'ajout d'un certain nombre de noeuds dans le réseau [58].

1.5.5 Sécurité

A cause de l'absence de câblage, les réseaux de capteurs sont exposés comme tout autre réseau sans fil, aux risques de l'interception de données et au brouillage qui augmente les interférences dans le réseau. De plus, les noeuds capteurs étant autonomes, reconfigurables et suivant une infrastructure de communication ad hoc, leur sécurité est moins assurée à cause du manque d'un point d'accès qui offre un niveau de sécurité plus élevé. En plus des problèmes de sécurité rencontrés dans les réseaux ad hoc, les réseaux de capteurs rencontrent d'autres handicaps et représentent une proie facile aux intrus selon la nature de l'application déployée. Plus précisément, pour une grande partie d'applications souvent critiques, les noeuds capteurs sont déployés, d'une manière dense et dans des zones éloignées et hostiles.

1.5.6 Modèle de liens de communication

Dans un réseau de capteurs, les communications entre les noeuds sont réalisées avec des liens sans fil à l'aide de media optique, infrarouge ou radio. La plupart des réseaux de capteurs utilisent des circuits de communication à radio fréquence vue leur faible coût ainsi que leur facilité d'installation [11].

Actuellement, un nombre important de travaux de recherche traitent la consommation énergétique des liens de communication de type radio. Différentes hypothèses concernant les caractéristiques radio, incluant la dissipation d'énergie dans les modes d'émission et réception de données, peuvent changer le comportement et les avantages des protocoles de routage. Beaucoup d'auteurs utilisent un modèle radio simple où la dissipation d'énergie est supposée égale à 50 nJ/bit, tandis que la meilleure technologie disponible offre une dissipation de 115 nJ/bit [43].

1.5.7 Coût de production

Comme la densité des réseaux de capteurs est généralement très élevée, le prix d'un seul capteur est très important pour définir le coût total du réseau. Les prix des noeuds capteurs varient selon, leur puissance, leurs capacités physiques et énergétiques et le nombre de capteurs qu'ils contiennent (on peut avoir plusieurs capteurs dans un seul noeud). Si le réseau est plus cher que le déploiement d'un ensemble de capteurs ordinaires, alors le coût du réseau n'est pas justifié. Autrement dit, si on arrive à accomplir les tâches désirées par un réseau de capteurs avec des noeuds ordinaires à prix raisonnables, alors l'utilisation des noeuds plus chers n'est pas vraiment efficace.

Les protocoles de routage doivent prendre en compte cette contrainte financière et tentent de trouver des techniques qui permettent l'utilisation des noeuds moins chers tout en garantissant une durée de vie du réseau élevée.

Toutes ces vulnérabilités transforment certains problèmes ayant des solutions évidentes dans les réseaux sans fil classiques, en des problèmes complexes et difficiles à résoudre dans les réseaux de capteurs. Ces défis devraient gagner une attention importante dans la communauté des chercheurs pour que des solutions et des mécanismes de communication efficaces puissent être réalisés. Ainsi, un nombre important de protocoles de routage ont été conçus pour les réseaux de capteurs. Chacun possède ses propres propriétés selon les caractéristiques étudiées et les métriques de routage considérées.

1.6 Métriques de routage et durée de vie du réseau

Les métriques de routage sont les critères de performances selon lesquels on mesure l'efficacité des protocoles de routage. Un calcul de métrique est un algorithme qui traite un coût associé à un certain chemin de routage. Les protocoles de routage permettent aux noeuds de comparer les métriques calculées afin de déterminer les routes optimales à emprunter. Plus la métrique est optimale en parcourant un certain chemin, plus le protocole de routage considère que la probabilité d'atteindre la station de base à travers ce chemin est grande [12]. Plusieurs métriques peuvent affecter le routage en termes d'énergie, délai, débit, distance, perte de paquets, etc.

Les caractéristiques particulières des réseaux de capteurs modifient les critères de performances par rapport aux réseaux sans fil traditionnels. Dans les réseaux locaux filaires, les réseaux cellulaires ou les réseaux ad hoc, les critères les plus pertinents sont le débit, la latence et la qualité de service. En effet, les nouvelles activités telles que le transfert d'images, le transfert de vidéos et la navigation sur Internet requièrent un débit important, une faible latence et une bonne qualité de service. En revanche, dans les réseaux de capteurs, la longévité du réseau est fondamentale. D'une part, les réseaux de capteurs sont contraints en ressources énergétiques (voir section 1.5.1). D'autre part, ce type de réseaux trouve des applications ne nécessitant pas un débit élevé telles que la domotique, l'agriculture intelligente, l'environnemental, la santé, la sécurité, l'écologie, etc. [11]. De ce fait, la conservation de l'énergie est devenue un critère de performance prépondérant dans les réseaux de capteurs et se pose en premier lieu tandis que les autres critères comme le débit ou l'utilisation de la bande passante sont devenus secondaires [13].

En conséquence, la durée de vie du réseau est sans doute la métrique la plus importante dans l'évaluation de performances des protocoles de routage conçus pour les réseaux de capteurs. Elle correspond à la période de temps durant laquelle le réseau peut accomplir les tâches que lui sont affectées. Il existe différentes définitions de la durée de vie d'un réseau de capteurs [40]. Elle peut être définie comme étant la durée jusqu'à ce qu'un seul noeud capteur soit épuisé. Elle peut également être définie par le temps jusqu'au moment où une proportion de noeuds soient épuisés. Comme il peut s'agir de la durée jusqu'à ce que tous les noeuds capteurs soient épuisés. De toute évidence, plus ces durées sont longues, meilleur est le fonctionnement du réseau.

1.7 Conclusion

L'évolution rapide qu'a connue la technologie sans fil, a permis l'apparition des réseaux de capteurs ; de nouveaux systèmes de communication qui offrent plus d'avantages par rapport aux systèmes classiques. Cependant, de nouveaux challenges peuvent apparaître causés par leurs propriétés telles que : les limitations de ressources énergétiques et physiques, la densité du déploiement, la tolérance aux pannes, etc. Ces limitations influencent sur la conception des protocoles de routage au sein des réseaux de capteurs où l'utilisation des protocoles de routage conçus pour les réseaux ad hoc traditionnels est inappropriée. Ceci est en raison des caractéristiques par lesquelles se distinguent les deux types de réseaux. En conséquence, un nombre important de protocoles de routage ont été conçus pour les réseaux de capteurs selon les caractéristiques étudiées et les critères de performances considérés.

Malgré les progrès qui ont été faits, la consommation d'énergie continue d'être un critère de performance majeur et un facteur clé, exigeant davantage de recherches sur l'efficacité énergétique des protocoles de routage. Pour conclure, les protocoles de routage conçus pour les réseaux de capteurs ne peuvent pas se passer des métriques qui minimisent la consommation d'énergie pendant le routage.

Le prochain chapitre sera consacré pour l'étude d'un ensemble de techniques de conservation d'énergie au cours du routage qui constituent l'un des principaux domaines de recherche dans les réseaux de capteurs.

Chapitre 2

Réseaux de Capteurs Sans Fil : Routage, Conservation d'énergie, Clustering

2.1 Introduction

Comme nous l'avons déjà souligné dans le premier chapitre, c'est la transmission de données qui se révèle extrêmement consommatrice par rapport aux tâches de captage et de traitement de données. Le principal objectif du routage consiste donc à minimiser la consommation d'énergie tout en assurant que le réseau effectue ses tâches dans de meilleures conditions. Cette consommation peut être optimisée en réduisant le nombre de messages en transit ainsi que les distances entre les noeuds communicants.

Pour ce faire, beaucoup de travaux de recherche ont proposé des mécanismes de conservation d'énergie, telles les techniques d'agrégation de données, les communications multi-sauts, ainsi que d'autres techniques de routage que nous retrouverons dans la première section de ce chapitre. En second lieu, nous citerons certains algorithmes de routage orientés-énergie tel que LEACH qui est considéré comme étant le premier protocole de routage hiérarchique dans les réseaux de capteurs, et pour terminer, nous expliquerons la nécessité d'utilisation des modèles formels pour la vérification et l'évaluation des performances des protocoles de routage conçus pour les réseaux de capteurs.

2.2 Techniques de conservation d'énergie

Dans cette partie, nous présentons un panorama de techniques de conservation d'énergie durant le routage. Nous pouvons citer : la réduction des distances de transmission par des communications multi-sauts, l'agrégation de données, et les techniques de clustering [40].

2.2.1 Les communications multi-sauts

L'un des paradigmes de communication utilisé par les protocoles de routage consiste en des transmissions directes (figure 2.1), c'est-à-dire, chaque noeud envoie directement ses données à la station de base. Vu que la station de base dans un réseau de capteurs est située loin du champ de captage, et donc, loin des noeuds capteurs, ce mode de communication requiert une grande puissance de transmission sur chaque noeud capteur. Cela va entraîner une dissipation excessive de l'énergie des noeuds capteurs, ce qui induit à une diminution significative de la durée de vie du réseau.

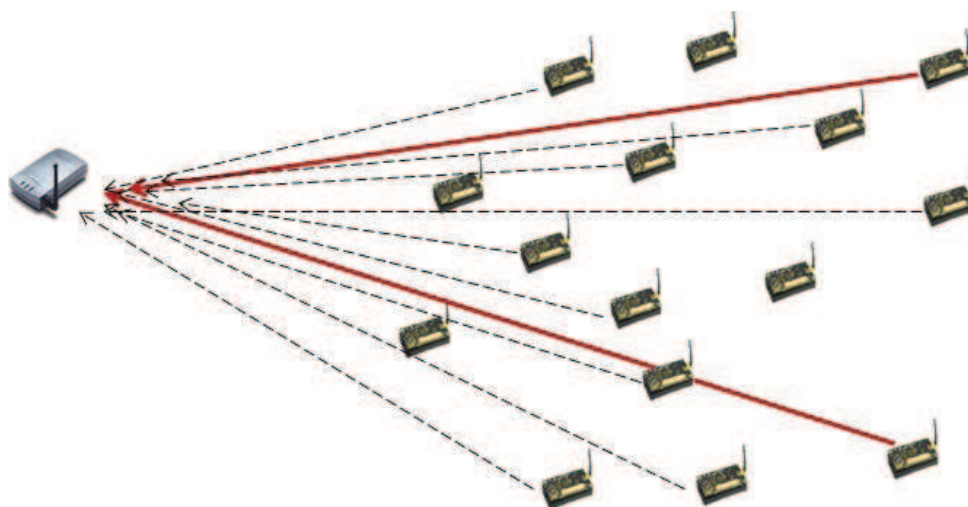


FIGURE 2.1 – Transmissions directes.

C'est pourquoi, les réseaux de capteurs adoptent des communications dites multi-sauts (*multi-hop*) (figure 2.2). Cela signifie que si la communication porte sur une distance importante, alors les données transmises seront relayées de noeud en noeud avant d'atteindre la destination. Cette technique part du principe qu'une transmission de données sur une longue distance est plus coûteuse énergiquement parlant que les transmissions réalisées sur des distances plus courtes. Donc, la communication avec des noeuds voisins est moins chère que la communication avec des noeuds qui sont situés à une plus grande distance (y compris la station de base). En conséquence, si les noeuds communiquent seulement avec leurs voisins, la communication sera moins chère, d'où l'énergie sera moins dissipée.

Dans ce paradigme, plusieurs routes peuvent être établies à partir d'une source vers une destination pour augmenter la tolérance aux pannes. Cette technique, appelée multi-chemins (*multi-paths*), sélectionne la route qui minimise au mieux l'énergie dissipée.

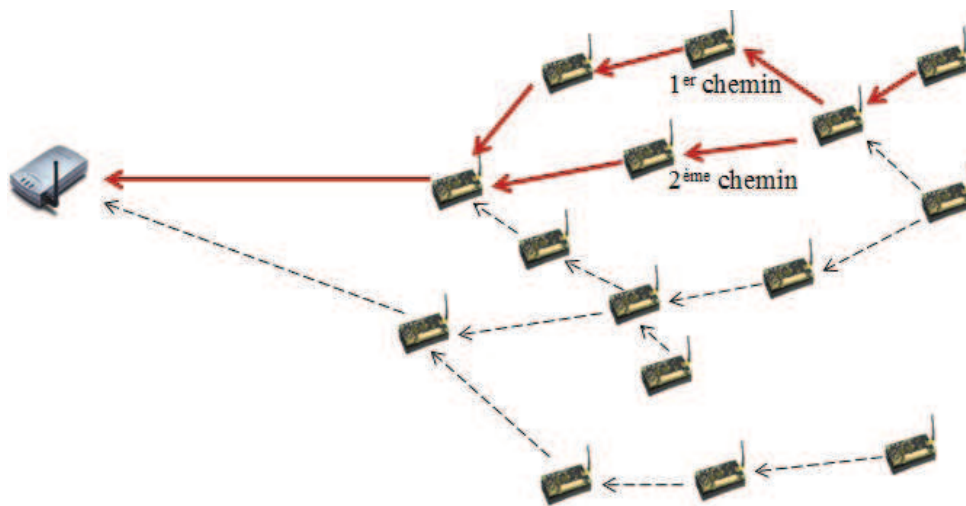


FIGURE 2.2 – Communications multi-sauts [18].

2.2.2 L'agrégation de données

Minimiser le flux de données dans un réseau de capteurs peut avoir un bon impact sur la consommation d'énergie. L'idée de base consiste à établir des routes liant plusieurs sources à une seule destination pour éliminer des données similaires et combiner les messages provenant de différents noeuds sources. Autrement dit, les noeuds intermédiaires rassemblent des données au fur et à mesure de leur acheminement vers la station de base. En effet, les réseaux de capteurs sont denses, ce qui implique que deux noeuds voisins peuvent capter la même information. Ainsi, il peut s'avérer judicieux d'éliminer les redondances et de réduire le taux de transmissions dans le réseau. La quantité de données est donc réduite tout en parcourant le réseau, d'où sa durée de vie est prolongée.

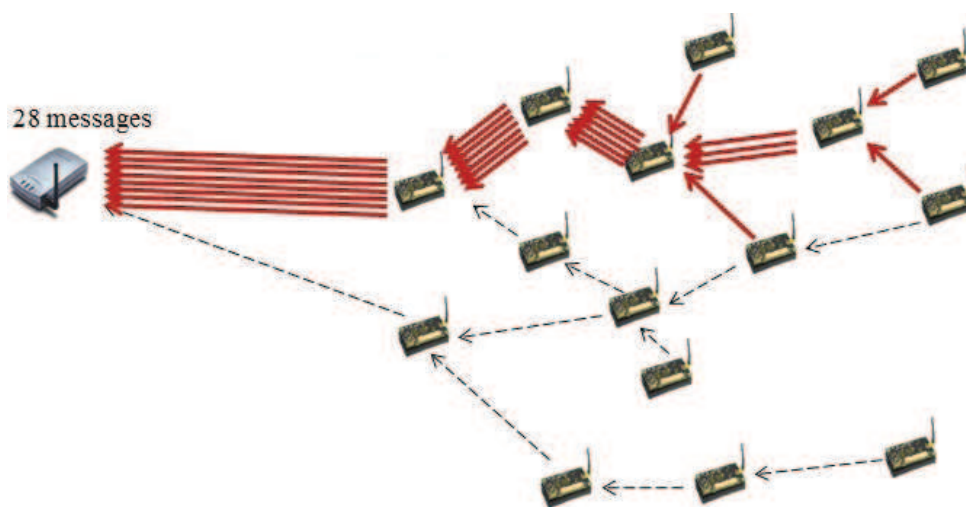


FIGURE 2.3 – Transmission sans agrégation de données [18].

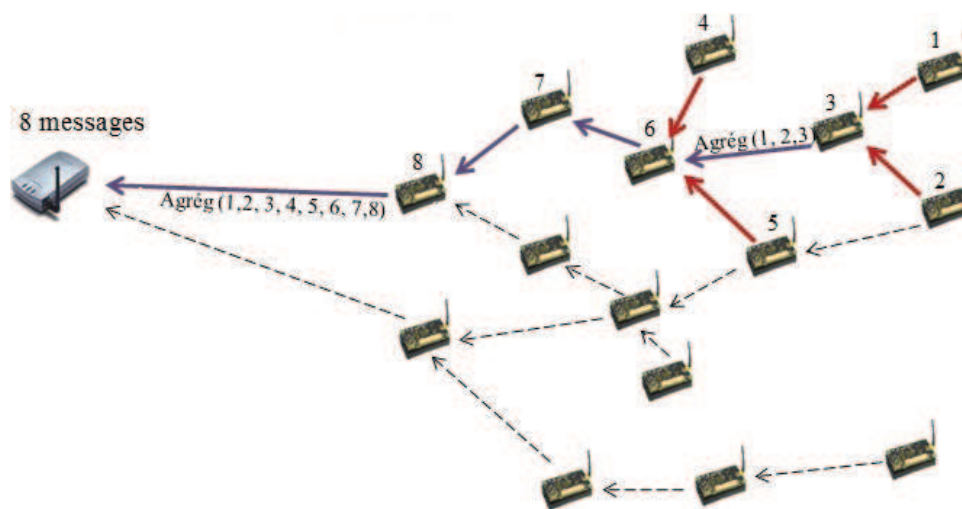


FIGURE 2.4 – Transmission avec agrégation de données [18].

La forme la plus simple que peut prendre une fonction d'agrégation est la suppression des messages dupliqués. Mais elle peut également être une fonction minimum ou maximum ou n'importe quelle fonction à plusieurs entrées.

Dans la figure 2.3, les paquets traversent le réseau saut-par-saut jusqu'à ce qu'ils atteignent la station de base, et chaque noeud intermédiaire prend le relais de ces paquets. Dans ce cas, un noeud intermédiaire envoie ses propres données et transmet les paquets reçus au prochain noeud tels quels, même s'ils sont dupliqués. Cela induit à une implosion de données et une surcharge de liens de communication. L'agrégation de données remédie à ce type de désagrément. Par exemple, dans la figure 2.4, le 3^{ème} noeud fusionne ses propres paquets avec les paquets provenant du premier et second noeud. Cette opération est répétée par les autres noeuds intermédiaires jusqu'à ce que le paquet agrégé arrive à la station de base [18].

2.2.3 Le clustering

Une autre méthode qui minimise la consommation d'énergie est la technique du *clustering* qui a été proposée comme un nouveau paradigme utile pour le routage dans les réseaux de capteurs. Elle englobe la technique de réduction des distances entre les noeuds communicants et la technique d'agrégation de données. Elle est définie par la hiérarchisation de l'architecture de communication du réseau. Les noeuds capteurs sont organisés en groupes disjoints ou *clusters*, identifiés par un seul noeud particulier appelé chef de groupe ou *Cluster Head* (CH). Les autres noeuds du groupe agissent comme des noeuds membres [40].

Durant la transmission, seuls les CH communiquent avec la station de base, et donc, seuls les CH adoptent des transmissions à longues distances. Dans le but de réduire la quantité d'informations transmise à la station de base, les CH agrègent les données capturées par les noeuds membres qui appartiennent à leur propre cluster, et envoient un paquet agrégé à la station de base. Dans ce cas, le routage devient plus simple puisqu'il s'agit de passer par les CH pour atteindre la station de base [37].

Dans de nombreux scénarios, le clustering permet une meilleure consommation d'énergie au sein du réseau par rapport aux communications directes et aux communications multi-sauts [9]. Dans une architecture à communications directes (figure 2.5(a)), les noeuds capteurs éloignés vont plus rapidement manquer d'énergie. Les régions les plus éloignées de la station de base seront donc non couvertes au bout d'un temps très court. En contrepartie, dans le cas d'une transmission multi-sauts (figure 2.5(b)), les noeuds sont homogènes (c'est-à-dire, ils effectuent les mêmes tâches de captage, de transmission, d'agrégation, etc.) et il n'y a que la station de base qui est chargée de la récolte d'informations, et, ces informations passent forcément par les noeuds qui l'entourent. Par conséquent, les noeuds proches de la station de base vont être rapidement épuisés car ils sont plus sollicités pour relayer les messages des autres noeuds capteurs. Ces noeuds sont appelés trous noirs ou *hotspots*.

Dans une architecture hiérarchique (figure 2.5(c)), les données des noeuds lointains seront envoyées à leurs CH mais pas directement à la station de base. Cela permet d'éviter l'isolement des régions éloignées. Par ailleurs, les données d'un cluster vont être combinées par le CH avant leur arrivée à la station de base ce qui allège le travail des noeuds qui l'entourent. De plus, contrairement aux autres architectures de communication, un réseau hiérarchique possède une forte scalabilité. En effet, l'ajout des noeuds ne dégrade pas les performances du réseau car le réseau peut gérer les nouveaux noeuds indépendamment des autres noeuds du réseau (par exemple, en les groupant et les associant à un CH sans affecter tous les noeuds du réseau).

Toutefois, chaque CH est responsable d'effectuer les tâches les plus coûteuses en termes d'énergie (les CH s'occupent de la coordination au sein de leurs clusters, de la gestion et des transmissions des données récoltées par les noeuds membres) afin d'alléger la charge sur les noeuds plus contraints en ressources énergétiques qui se consacrent uniquement au captage. Ainsi, lorsque le clustering est **statique**, c'est-à-dire, lorsque les CH ne changent pas de rôle durant toute la durée de vie du réseau, ils vont être rapidement épuisés. Par conséquent, le réseau va être divisé ce qui implique le découpage du réseau en secteurs

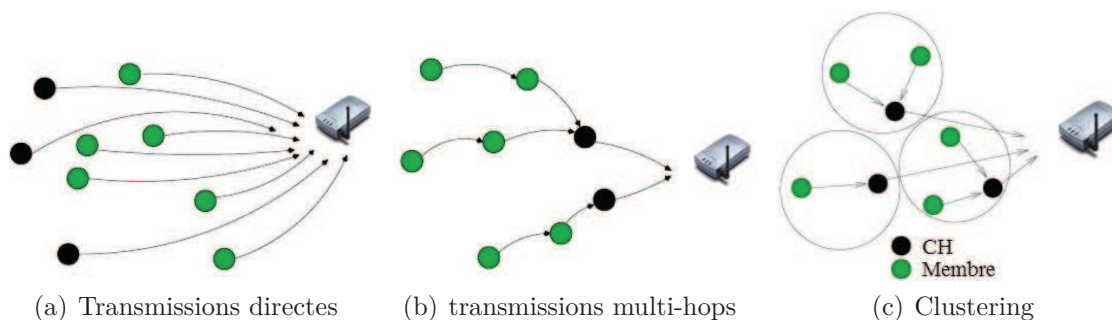


FIGURE 2.5 – Architectures de communication adoptées par les réseaux de capteurs [40].

inaccessibles. On peut alors penser à attribuer le rôle de CH aux différents noeuds à tour de rôle pour garantir une dissipation équitable d'énergie entre les noeuds. On obtient ainsi un clustering **dynamique**, selon lequel aucun noeud capteur ne sera très tôt épuisé par rapport aux autres.

Le premier protocole de routage qui a adopté le clustering dynamique est LEACH (*Low-Energy Adaptive Clustering Hierarchy*) [20]. C'est un protocole sur lequel est basée la conception d'une multitude de protocoles de routage hiérarchiques. TEEN (*Threshold sensitive Energy Efficient sensor Network protocol*) [59], PEGASIS (*Power Efficient Gathering in Sensor Information Systems*) [78] et EEDMC (*Energy-Efficient Distributed Multi-level Clustering Algorithm*) [53] étant les protocoles de routage les plus populaires. Chacun apporte une amélioration à LEACH, comme ; les critères d'élection des CH, les communications entre les CH, les communications entre les membres d'un cluster, etc.

2.3 Protocoles de routage hiérarchiques

Au cours de la dernière décennie, beaucoup de travaux de recherche essaient de perfectionner les protocoles hiérarchiques afin de fournir des solutions aux défis mentionnés en section 1.5. L'un des protocoles hiérarchiques les plus fréquemment utilisés dans les réseaux de capteurs est le protocole LEACH. Ce protocole est basé sur un ensemble de suppositions que l'on cite dans les points suivants :

- Les liens de communication utilisent le modèle radio pour estimer l'énergie dissipée dans les transmissions.
- La station de base est fixe, puissante et située loin des autres noeuds capteurs.
- Tous les noeuds sont initialement homogènes et commencent avec la même quantité d'énergie et meurent approximativement en même temps.
- Tous les noeuds sont considérés statiques, homogènes, et les canaux de communica-

tion radios sont supposés symétriques.

- Ces noeuds sont prévus pour capter leur environnement d'une manière continue, et ont, donc, un taux de données fixe à transmettre. Cela revient à dire que LEACH est conçu pour la classe des réseaux de collection des données d'environnement. Cette hypothèse n'est pas conforme au cas des réseaux de surveillance et de sécurité, ni au cas des réseaux de poursuite.

L'idée de base consiste à former des clusters où les CH demeurent ainsi pour une période de temps appelée round, puis, ils changent de rôles durant d'autres rounds afin d'avoir une dissipation d'énergie équitable au sein du réseau. Au début de chaque round, chaque noeud détermine la possibilité d'être un CH pendant la période en cours. S'il décide de l'être, il annonce sa décision à tous les noeuds voisins. Les noeuds non-CH se joignent à l'un des CH élus et qui possède une plus grande force du signal (donc le plus proche). Une fois les clusters formés, les CH affectent des slots de temps à leurs membres. Chaque membre capte les informations de son environnement et les envoie à son CH durant son propre slot. Cela permet d'éviter le problème de collision. Utilisés comme passerelle pour atteindre la station de base, les CH agrègent les données reçues et envoient le résultat final à la station de base.

Le protocole TEEN est similaire au protocole LEACH, sauf que les noeuds ne sont pas supposés avoir un taux fixe de données à transmettre. Ainsi, TEEN convient pour les applications critiques qui englobent les réseaux de surveillance et de sécurité. L'architecture du réseau de capteurs selon TEEN est basée sur la formation des clusters selon la même stratégie que le protocole LEACH. La différence se situe au niveau de la transmission de données. Durant cette étape, TEEN utilise deux paramètres définis par l'utilisateur : le paramètre *ht* (*hard threshold*) et le paramètre *st* (*soft threshold*), et ce, pour pouvoir déterminer si la donnée captée sera transmise ou non à la station de base. Si la valeur captée par un membre excède le paramètre *ht* pour la première fois, elle est stockée dans une variable, et puis, elle est transmise durant son propre slot de temps. Si la valeur captée dépasse, par la suite, la valeur stockée par une magnitude *st*, le noeud décide de la transmettre et stocke cette nouvelle valeur dans son cache pour des comparaisons ultérieures.

PEGASIS est une version améliorée du protocole LEACH. Ce protocole part des mêmes hypothèses que le protocole LEACH. L'idée de base est de former des chaînes au sein des clusters, c'est-à-dire, les membres d'un cluster utilisent des communications multi-sauts afin que chaque noeud transmette et reçoive uniquement des données de son plus proche voisin. Ces données seront agrégées jusqu'à ce qu'elles atteignent un seul noeud

appelée *Leader*. Ce dernier est sélectionné à partir de cette chaîne pour transmettre les données finales à la station de base. Chaque noeud de la chaîne joue le rôle de *Leader* au cours d'un round et le rôle de membre durant d'autres rounds, ce qui est similaire au protocole LEACH.

EEDMC est un autre protocole de routage hiérarchique qui prend profit des protocoles LEACH et PEGASIS. Il a été récemment proposé dans [53]. Le facteur clé est qu'un noeud capteur, ayant une plus grande énergie résiduelle et étant plus proche de la station de base, convient mieux pour jouer le rôle d'un CH. La principale contribution de ce travail est que, dans chaque round, le choix des CH est basé sur la plus grande proportion entre l'énergie résiduelle de chaque noeud et la moyenne des énergies résiduelles des noeuds voisins. Après la formation des clusters, une chaîne entre les CH est construite suivant le paradigme de communication multi-sauts. Chaque CH communique seulement avec son plus proche CH et le CH le plus proche de la station de base sera sélectionné comme *Leader* de cette chaîne (les auteurs ont adopté le même principe des chaînes du protocole PEGASIS).

Après cette brève présentation, nous détaillerons dans la prochaine section, l'algorithme du protocole LEACH, étant donné qu'il est considéré comme l'ancêtre des protocoles de routage hiérarchiques dans les réseaux de capteurs.

2.4 Protocole de routage LEACH

Dans cette partie, nous décrivons les différentes étapes impliquées dans le fonctionnement du LEACH. Par la suite, nous expliquons le modèle radio utilisé par ce protocole pour calculer l'énergie consommée par les noeuds du réseau.

2.4.1 Fonctionnement du protocole LEACH

Avant d'exécuter l'algorithme du protocole LEACH, on désire avoir un certain nombre de CH. Ce nombre, que l'on note K , est fixe et il est inchangé durant toute l'exécution. On estime que le pourcentage optimal du nombre de CH désirés devrait être de 5% à 15% [20] du nombre total de noeuds que l'on note N . Si ce pourcentage n'est pas respecté, cela mènera à une grande dissipation d'énergie dans le réseau. En effet, si le nombre de CH est très élevé, on aura un nombre important de noeuds (CH) qui se consacrent aux tâches très coûteuses en ressources énergétiques (on tend vers le problème de la communication directe). Ainsi, on aura une dissipation d'énergie considérable dans le réseau. De plus, si le nombre de CH est très petit, ces derniers vont gérer des groupes de grandes tailles.

Ainsi, ces CH s'épuiseront rapidement à cause du travail important qui leur est demandé.

L'algorithme se déroule en *rounds* qui ont approximativement le même intervalle de temps déterminé au préalable. Chaque round est constitué d'une phase d'initialisation et d'une phase de transmission.

2.4.1.1 Phase d'initialisation

Au début de chaque round, chaque noeud capteur détermine la possibilité d'être un CH pendant le round en cours qui commence à l'instant t . Pour ce faire, chaque noeud génère un nombre aléatoire entre 0 et 1 et le compare à un seuil $T_i(t)$ calculé en fonction de K , N et du round courant : c'est la proportion entre le nombre de CH désirés et le nombre de noeuds qui n'ont pas encore été élus CH durant les r rounds précédents.

$$T_i(t) = \frac{K}{N - K \times (r \bmod \frac{1}{K})} \quad [20]$$

Si ce nombre aléatoire est inférieur à $T_i(t)$, le noeud décide d'être CH et annonce sa décision à tous les noeuds voisins en leur diffusant un message d'avertissement ADV. Les autres noeuds qui décident de ne pas être CH, deviennent des membres et se joignent à l'un des CH élus après avoir reçu leurs décisions. La sélection du CH adéquat est basée sur l'amplitude du signal reçu ; le CH ayant le signal le plus fort (i.e. le plus proche) sera choisi. En cas d'égalité des signaux, les noeuds non-CH choisissent alors aléatoirement leur CH. Chaque membre informe son CH de sa décision. Une fois que le CH a reçu les demandes d'appartenance, il envoie à ses membres des messages d'acquiescement JOIN-REQ. Après la formation des groupes, chaque CH agit comme un centre de commande local pour coordonner les transmissions des données au sein de son cluster. Il crée un ordonnanceur TDMA¹ (*schedule TDMA*) et assigne à chaque membre un slot de temps durant lequel il peut transmettre ses données.

2.4.1.2 Phase de transmission

La phase de transmission est plus longue que la phase d'initialisation. Durant cette phase, les membres émettent leurs données captées pendant leurs propres slots afin d'éviter le problème de collision de paquets de données. De plus, cela leur permet d'éteindre leurs interfaces de communication en dehors de leurs slots afin d'économiser leur énergie. Ces données sont ensuite agrégées par les CH qui les fusionnent et les compressent, et, envoient le résultat final à la station de base.

1. Le schéma TDMA (*Time Division Multiple Access*) fait partie des protocoles MAC de la couche liaison de données. Il permet de diviser le temps en intervalles (*time-slot*) attribués à chaque noeud. Un noeud aura donc le droit d'accès au canal en utilisant toute la plage de la bande passante du canal, mais devra émettre ses données pendant seulement les intervalles de temps qui lui sont accordés.

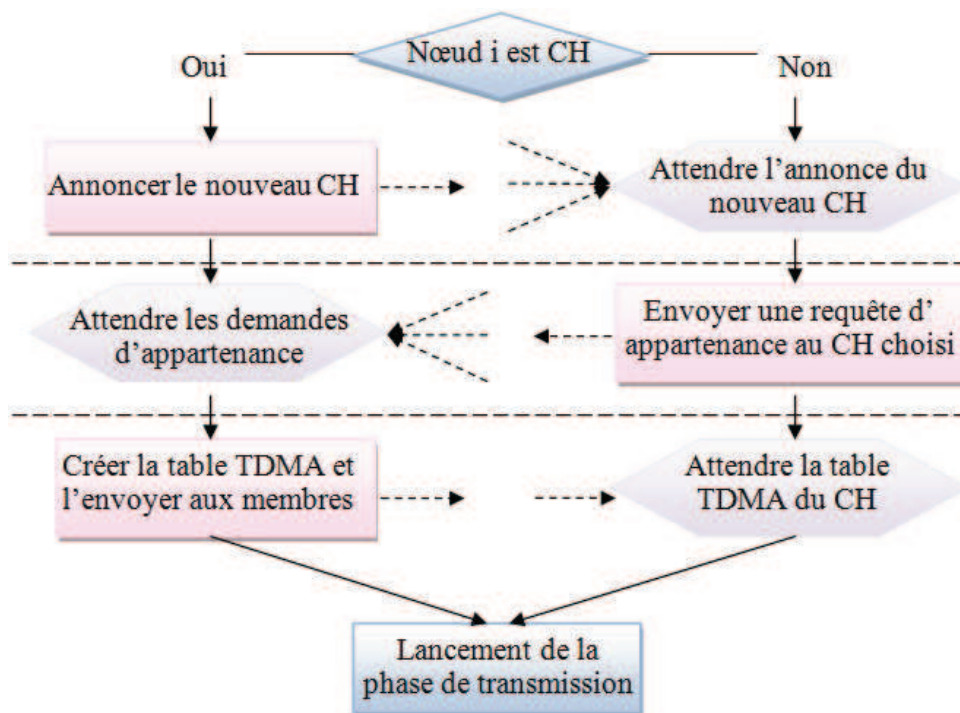


FIGURE 2.6 – Diagramme de la phase d'initialisation du protocole LEACH [12].

Après un certain temps prédéterminé, le réseau va passer à un nouveau round. Ce processus est répété jusqu'à ce que tous les nœuds du réseau seront élus CH, une seule fois, tout au long des rounds précédents. Dans ce cas, le round est réinitialisé à 0. Il faudra N/K rounds durant lesquels un nœud est élu seulement une seule fois en tant que CH avant que le round ne soit réinitialisé à 0. Ceci part du principe que seuls les nœuds qui n'ont pas encore été CH, ont vraisemblablement une énergie résiduelle suffisante par rapport les autres et ils pourront alors être choisis pour le round suivant.

2.4.2 Modèle radio du protocole LEACH

Un nœud capteur consomme l'énergie selon deux manières : l'énergie utilisée pour exécuter le circuit de communication électronique et l'énergie utilisée pour amplifier le signal d'émission (voir figure 2.7).

L'énergie utilisée pour exécuter le circuit de communication électronique dépend du nombre de bits dans le message. Soit k la longueur du message et E_{elec} l'énergie utilisée pour exécuter le circuit de communication électronique par bit. Ainsi, l'énergie utilisée pour exécuter le circuit de communication pour un message de k bits est $E_{elec} \times k$. Cette énergie est la même que ce soit pour l'émission ou pour la réception.

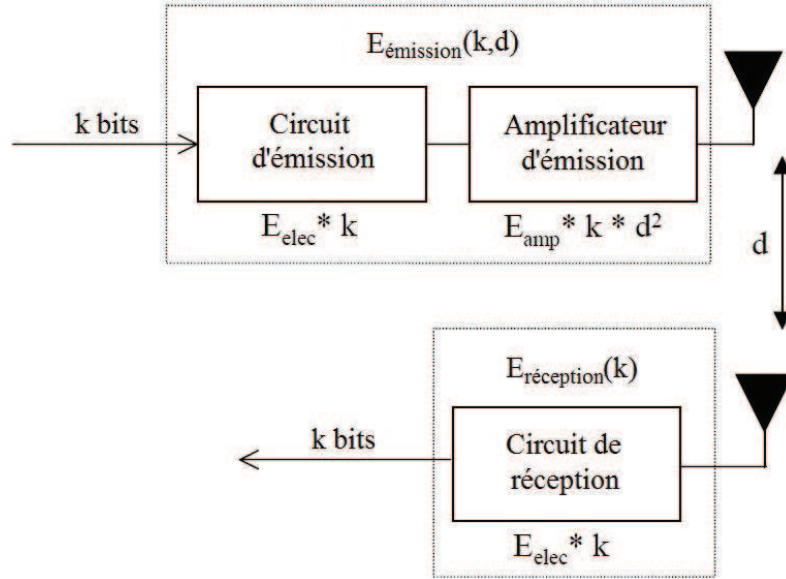


FIGURE 2.7 – Modèle de liens de communication [20].

Quant à l'énergie utilisée pour l'amplificateur d'émission, elle dépend du nombre de bits dans le message et de la distance entre l'émetteur et le récepteur. On définit E_{amp} comme étant l'énergie utilisée pour l'amplificateur d'émission par bit. Ainsi, l'énergie nécessaire pour amplifier le signal d'émission afin d'envoyer un message de k bits à travers une distance d est égale à $E_{amp} \times k \times d^2$.

En résumé, pour émettre un message de k bits à travers une distance d , l'énergie dissipée par la radio est :

$$E_{émission}(k, d) = E_{elec} \times k + E_{amp} \times k \times d^2$$

d'où,

$$E_{émission}(k, d) = k(E_{elec} \times + E_{amp} \times d^2)$$

Et pour recevoir ce message, l'énergie dissipée par la radio est :

$$E_{réception}(k) = E_{elec} \times k$$

Différents protocoles de routage utilisent ce modèle radio. Cela permet de pouvoir faire une comparaison entre l'énergie consommée par le protocole LEACH et par le protocole amélioré, et, évaluer leurs performances avec les mêmes paramètres énergétiques. Évaluer ces performances et effectuer une comparaison entre elles est dans le but de garantir l'efficacité de chaque amélioration du protocole de routage perfectionné. D'une manière générale, les protocoles de routage doivent être évalués avant leur mise en oeuvre afin de garantir leur efficacité et détecter les erreurs lors de la conception, et ce, dans le but d'arriver à une meilleure fiabilité avec moindre coût.

Dans ce qui suit, nous expliquerons la nécessité de l'évaluation de performances des protocoles de routage en présentant quelques méthodes d'évaluation utilisées dans les réseaux de capteurs [82]. Parmi ces méthodes, la simulation est la plus répandue. Beaucoup de simulateurs ont été conçus pour les réseaux de capteurs, comme par exemple, PowerTOSSIM qui prend en charge la consommation d'énergie. Même s'il existe des simulateurs spécifiques aux réseaux de capteurs, les simulateurs classiques sont parfois utilisés dans le contexte de ces réseaux (citons à titre d'exemple, le simulateur NS2). Nous présenterons ensuite la nécessité d'utiliser des méthodes d'analyses basées sur des modélisations mathématiques.

2.5 Nécessité de la vérification formelle

Dans cette section, nous expliquons la nécessité d'évaluer les performances d'un système, et puis, la nécessité d'avoir recours à des méthodes formelles pour évaluer ces performances.

2.5.1 Pourquoi évaluer les performances d'un système ?

L'évaluation des performances d'un système permet de s'assurer de son bon fonctionnement. D'une part, pour un grand système tel un réseau de capteurs, le nombre de noeuds qui le constituent, peut atteindre plusieurs milliers et entraîne donc un coût financier relativement important. Il faut donc réduire au maximum les erreurs de conception. D'autre part, lorsque le système n'existe pas encore, son analyse permet aux concepteurs de prévoir son fonctionnement. De plus, les expériences sur un système réel sont trop coûteuses en termes de ressources matérielles et humaines et elles ne sont pas représentatives de tous les environnements possibles. Dans ce cas, cette phase d'analyse et d'évaluation permet de caractériser son comportement global pour différents environnements réels [36].

2.5.2 De la simulation vers les modèles formels

L'approche traditionnelle de la conception d'un système consiste à le concevoir sur papier, à le réaliser et à faire des expériences sur le système correspondant à des scénarios types afin de vérifier si son comportement est satisfaisant et s'il est nécessaire de l'améliorer, ou même, le reconcevoir. Néanmoins, une telle méthode est longue et coûteuse, et dans certains cas, il est impossible de réaliser des expériences sur le système. De plus, lorsqu'on est face aux systèmes critiques, ces techniques de tests pourront être dangereuses si des

informations manquent sur leur comportement.

Ainsi, il a fallu penser à définir le système à l'aide d'un modèle qui permet en général de calculer numériquement les résultats, ce qui est plus court et plus économique que de les mesurer au cours d'une expérience. Cela est désigné par le terme de simulation. Celle-ci permet ainsi dans une certaine mesure de tester si le comportement du système est satisfaisant. Elle ne remplace pas complètement l'expérimentation si à l'issue de la simulation le comportement du système semble satisfaisant. En effet, le problème de la simulation est que, même si le modèle est bon, on ne peut tester qu'un nombre limité de scénarios. Si le choix des scénarios types n'est pas pertinent par rapport à l'ensemble des scénarios auxquels sera confronté le système durant son existence, il est difficile de prévoir si le comportement du système sera satisfaisant dans tous les cas [82].

Une autre approche consiste à garantir qu'à partir d'une propriété du modèle, le comportement du système sera satisfaisant ou non pour une famille de scénarios. On parle alors d'analyse du système qui repose sur l'étude des propriétés mathématiques de son modèle [27]. Ces modèles mathématiques doivent être facilement analysables. Une fois le modèle mathématique proposé, de nombreuses analyses sont possibles. Ces analyses peuvent s'intéresser au cas moyen comme le font les simulations mais elles permettent aussi d'exhiber le pire cas du modèle. Ces modèles permettent ainsi une meilleure compréhension du système.

2.6 Conclusion

Dans ce chapitre, nous avons étudié un certain nombre de techniques qui minimisent la consommation d'énergie durant le routage. Nous avons cité trois méthodes : les communications multi-sauts, l'agrégation de données et la technique du clustering. L'objectif principal du clustering est d'économiser l'énergie de noeuds capteurs étant donné que les communications multi-sauts sont adoptées au sein des clusters et les transmissions à longues distances sont uniquement assurées par les CH plutôt que par tous les noeuds du réseau, en plus de l'agrégation des données qui est effectuée par les CH afin de diminuer le nombre de messages transmis à la station de base.

Dans le cas où les CH maintiennent leur rang, ils s'épuiseront rapidement ce qui implique un isolement de certaines régions du champ de captage. C'est pourquoi, il a fallu penser à changer les rangs des noeuds à tour de rôle afin de réaliser une dissipation d'énergie équitable au sein du réseau. LEACH étant le premier protocole qui adopte ce type

de clustering dynamique, réalise une réduction significative de la dissipation d'énergie et peut étendre la durée de vie du réseau par un facteur de 8 [20], comparé aux autres protocoles basés sur le routage multi-sauts ou le clustering statique. L'idée proposée par LEACH a été une inspiration pour de nombreux protocoles de routage hiérarchiques, bien que certains protocoles aient été développés de manière indépendante.

Afin de s'assurer de leur bon fonctionnement, les protocoles de routage doivent passer par une phase préliminaire qui est la phase d'analyse et d'évaluation de performances. Pour ce faire, il est possible de procéder à des bandes d'essais une fois que le protocole a été réalisé. Mais dans ce cas, la correction des erreurs risque d'être extrêmement coûteuse. En conséquence, la communauté des chercheurs a pensé à des outils plus rapides et moins coûteux ; ce sont les simulateurs. Ceux-ci permettent l'analyse de gros réseaux, mais ils ne permettent pas l'étude exhaustive de toutes les exécutions du système. Toutefois, le besoin d'arriver à de bons résultats et avoir une plus grande fiabilité a permis d'introduire des modèles formels basés sur des principes mathématiques solides. Nombreux sont ces modèles proposés pour modéliser différents concepts dans les réseaux de capteurs. Le chapitre suivant sera donc consacré à l'étude d'un ensemble de ces outils formels pour les réseaux de capteurs.

Chapitre 3

Réseaux de Capteurs Sans Fil : Modélisation Formelle

3.1 Introduction

De nombreux systèmes informatiques existants, à différents niveaux de complexité et d'utilisation, font apparaître certaines défaillances ou dysfonctionnements. Dans certains cas, ces erreurs de fonctionnement n'ont pas d'impact grave sur l'environnement qui les entourent. D'autres peuvent, par contre, avoir des répercussions néfastes dans des applications jugées critiques [19]. Il est nécessaire alors d'analyser et d'évaluer les performances de tels systèmes de manière formelle, en essayant de penser à tous les scénarios possibles que peut rencontrer un système et traiter ceux qui pourraient mener à un dysfonctionnement en décelant les événements qui ne devraient pas se produire a priori.

Dans notre travail, nous nous sommes focalisés sur les réseaux de capteurs sans fil pour lesquels plusieurs modèles formels ont été proposés pour leur description et leur analyse. Dans ce chapitre, nous présentons d'abord les caractéristiques que doivent avoir les modèles formels. Nous étudions, par la suite, un ensemble de ces formalismes en citant quelques travaux existants de modélisation formelle de réseaux de capteurs. Pour conclure, nous donnons une synthèse sur ces différentes approches.

3.2 Caractéristiques d'un modèle formel

L'utilisation des modèles formels prend une part de plus en plus importante dans les projets informatiques, que cela soit pour la définition des systèmes, leur conception, leur réalisation ou leur mise en oeuvre, leur maintenance et leur exploitation. En effet, ils permettent de décrire de façon non équivoque un système informatique.

Par définition, un modèle est une représentation d'un système, souvent en termes mathématiques, des relations qui existent entre les sorties et les entrées de ce système. Un modèle serait idéal, s'il est capable de reproduire de façon exacte le comportement du système. Cependant, il est bien connu que l'obtention d'un modèle idéal est impossible. Au mieux, on peut obtenir un modèle qui permet de déterminer de façon optimisée le comportement du système. En général, un modèle formel doit satisfaire quatre critères essentiels [30, 57]. Il doit être :

- Expressif : il doit avoir une puissance d'expression. Autrement dit, il doit décrire de manière assez complète les caractéristiques du système modélisé et il ne doit pas être limité par des impossibilités techniques.
- Simple : il doit permettre de modéliser simplement, sans contorsions inutiles, les caractéristiques essentielles du système modélisé.
Cette simplicité constitue avec l'expressivité deux notions rivales. Ainsi, il faut arriver au bon compromis entre les deux, c'est-à-dire, un modèle doit être suffisamment expressif pour décrire la réalité des systèmes tout en restant raisonnablement analysable.
- Universel : il doit être utilisable dans tous les domaines d'activité et ne pas être étroitement dépendant d'un domaine spécifique afin de permettre un gage de pérennité et des économies d'échelle dans le développement d'outils.
- Opérationnel : il doit avoir un caractère exécutable pour produire un code exécutable ou pour que les modélisations des systèmes puissent être traitées par des outils de simulation.

3.3 Modèles formels pour les réseaux de capteurs

Dans cette section, nous allons donner un aperçu sur un certain nombre de modèles formels utilisés pour analyser et évaluer les performances des réseaux de capteurs sans fil à savoir les réseaux de neurones, les chaînes de Markov, les algèbres de processus, les automates temporisés et les réseaux de Petri.

3.3.1 Modèles à base de réseaux de neurones

Les réseaux de neurones formels représentent un modèle mathématique dont la conception est très schématiquement inspirée du fonctionnement des neurones biologiques. Dans cette partie, nous donnons un aperçu sur ce type de modèles. Ensuite, nous présentons un certain nombre de travaux de recherche qui ont utilisé les réseaux de neurones pour la modélisation formelle des réseaux de capteurs. Nous terminons cette partie avec un exemple pratique de l'utilisation des réseaux de neurones dans les réseaux de capteurs.

3.3.1.1 Présentation des réseaux de neurones

S'inspirant du neurone biologique, Warren McCulloch et Walter Pitts ont proposé en 1943 un modèle mathématique et informatique simple sous l'appellation du neurone formel [76]. Ce dernier est un processeur très simple (simulé sur ordinateur ou réalisé sur circuit intégré) imitant grossièrement la structure et le fonctionnement d'un neurone biologique [80]. En effet, il reproduit certaines caractéristiques biologiques, plus précisément, les dendrites, l'axone et les synapses, au moyen de fonctions et de valeurs numériques. Il possède généralement plusieurs entrées et une sortie qui correspondent respectivement aux dendrites et à l'axone du neurone biologique. Les actions excitatrices des synapses sont représentées par des coefficients numériques associés aux entrées et intitulés poids synaptiques [89].

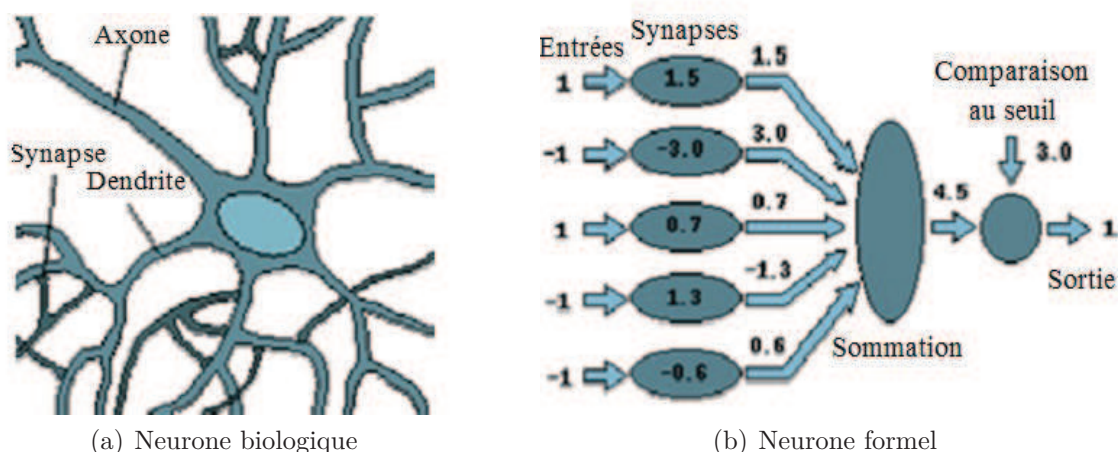


FIGURE 3.1 – Analogie neurone biologique et neurone formel.

Comme le montre la figure 3.1(b), il s'agit d'un automate binaire qui effectue une somme pondérée (potentiel) de ses entrées puis la compare à un seuil : si elle le dépasse, la sortie vaut 1 et le neurone est actif, sinon elle vaut 0 et le neurone est inactif. Il existe aujourd'hui d'autres types de neurones, mieux adaptés aux tâches de traitement du si-

gnal, dont la sortie n'est pas le signe de leur potentiel, mais une fonction f non linéaire dérivable de ce potentiel, telle une tangente hyperbolique. Cette fonction est dite fonction d'activation du neurone (voir figure 3.2) [80].

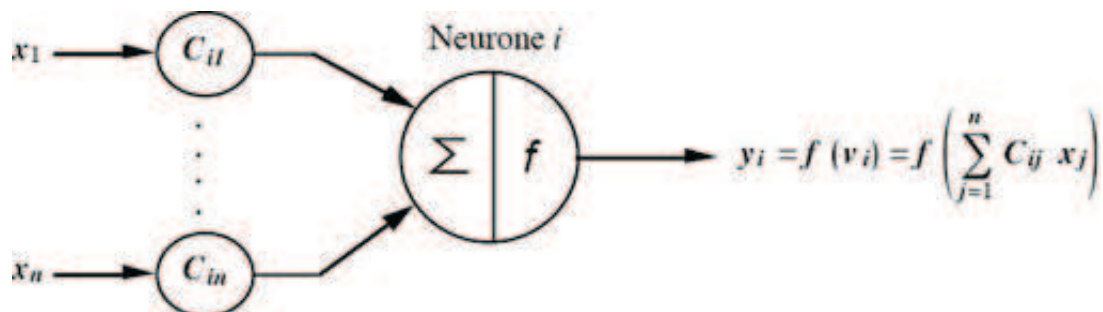


FIGURE 3.2 – Un neurone formel.

Le neurone formel est l'unité élémentaire des réseaux de neurones formels dans lesquels il est interconnecté à ses semblables pour calculer des fonctions arbitrairement complexes, utilisées pour diverses applications en statistiques, en automatique et en intelligence artificielle.

Il existe différents types de réseaux de neurones dont chacun possède des propriétés liées à l'application désirée, mais le plus courant est celui représenté sur la figure 3.3.

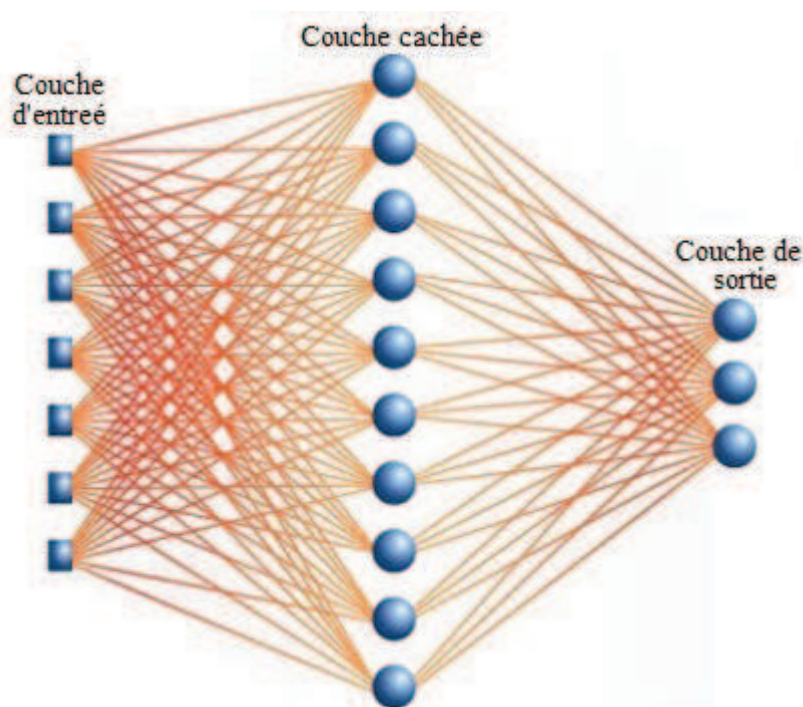


FIGURE 3.3 – Exemple d'un réseau de neurones formels.

Les réseaux de neurones sont essentiellement constitués d'une succession de couches, chacune est composée de noeuds de neurones et prend ses entrées sur les sorties de la précédente. La première couche est appelée couche d'entrée, et son rôle est d'accepter les données du problème. La couche du milieu est appelée couche cachée et permet de recevoir les données provenant de la couche précédente, et d'agir comme entrée pour la couche suivante. La dernière couche est appelée couche de sortie et sa fonction est de présenter les données de sortie après que leur traitement soit terminée.

Pour qu'un réseau de neurones puisse effectuer une tâche, il faut ajuster les valeurs de ses coefficients. C'est en variant les poids synaptiques, que les valeurs de sortie pourront être modérées. Grâce à cette approche, les données d'entrée sont remises à la couche d'entrée et la sortie est comparée avec la valeur désirée. Si elles diffèrent, les poids des couches intermédiaires sont légèrement modifiés pour réduire l'erreur de la sortie. Cette opération est répétée plusieurs fois et avec de nombreux ensembles de données, jusqu'à ce que le réseau obtienne le résultat correct pour toutes les valeurs d'entraînement qui lui ont été fournies. La procédure d'ajustement de ces coefficients est appelée apprentissage [15].

3.3.1.2 Réseaux de neurones pour les réseaux de capteurs

Récemment, on s'intéresse fortement à l'utilisation des outils intelligents basés sur les réseaux de neurones pour modéliser certains concepts des réseaux de capteurs sans fil. Plusieurs modèles ont été implémentés tels que *Multi-Layer Perceptrons* (MLPs), *Self-Organizing Maps* (SOMs), *Adaptive Resonance Theory* (ART) et ses variantes (ART1 et *Fuzzy ART*).

Les réseaux de neurones sont généralement utilisés dans les réseaux de capteurs dans le but d'avoir une meilleure précision, fiabilité et qualité d'informations de l'environnement avec moins de dissipation énergétique. Des techniques de classification ont été utilisées pour organiser les données captées afin d'améliorer la gestion de l'énergie au niveau de chaque noeud capteur et de réduire sa consommation pendant le routage (vu que les communications sont les plus consommatrices en terme d'énergie). D'autres techniques de classification ont été utilisées pour optimiser la synchronisation et maintenir l'ordre des paquets routés entre les noeuds capteurs [15].

Dans [65], les auteurs ont introduit la notion des réseaux de capteurs de neurones sans fil. Ce sont des réseaux utilisés pour des applications jugées intelligentes (*smart applications*) qui offrent une meilleure scalabilité et une plus grande fiabilité et précision dans la surveillance de l'habitat, l'amélioration de la qualité de vie, la sécurité domestique, la sécurité des patients, etc. Pour ce faire, les auteurs ont comparé un noeud capteur à un noeud formel. Une analogie entre un neurone formel et un noeud capteur est donnée sur la figure 3.4.

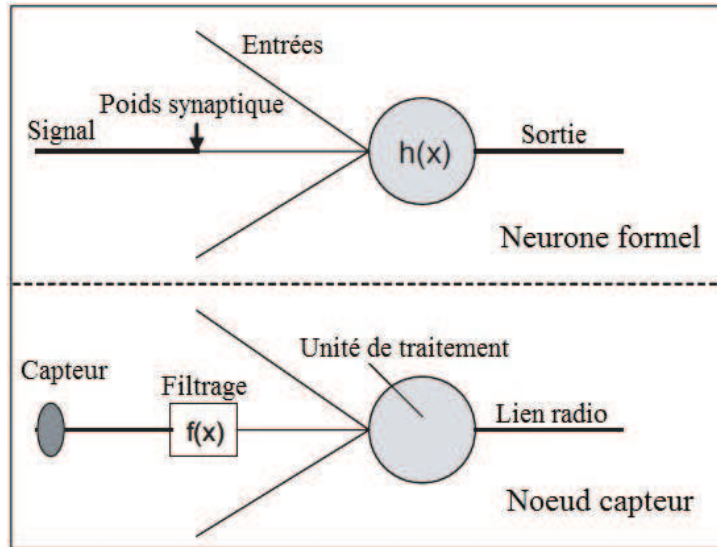


FIGURE 3.4 – Noeud capteur vs neurone formel. [65]

Dans un neurone formel, les signaux d'entrées sont traités et ajustés par des poids synaptiques. La fonction $h(x)$ est la fonction d'activation du neurone, elle produit une sortie à partir des valeurs données en entrée. De même, un noeud capteur surveille l'environnement et convertit les grandeurs physiques captées en des signaux électriques qui sont par la suite filtrés et traités par l'unité de traitement. Le résultat est transmis via un lien radio. Cette forte analogie montre que le noeud capteur lui-même peut être vu comme un neurone formel. Ainsi, on peut passer des réseaux de capteurs ordinaires vers des réseaux de capteurs de neurones dont chaque noeud capteur peut lui-même exécuter un réseau de neurones pour décider de l'action de sortie. Les auteurs ont prouvé que l'implémentation d'un réseau de neurones sur chaque noeud capteur est bénéfique vu que les caractéristiques des réseaux de capteurs (comme le parallélisme, les limitations physiques et énergétiques) sont considérées par les réseaux de neurones. De plus, les réseaux de neurones offrent une qualité de service considérable, parce qu'ils permettent la prédiction et la correction d'erreurs de transmission (comme les pertes de paquets) causées par les endroits de déploiement hostiles qui rendent les noeuds capteurs et les liens de communication souvent très vulnérables.

Dans [46], les auteurs ont utilisé les réseaux de neurones pour améliorer la fiabilité de l'information de l'environnement. Ils ont utilisé une architecture hiérarchique, et seuls les noeuds membres sont munis d'un réseau de neurones (on n'implémente donc pas un réseau de neurones sur tous les noeuds du réseau). L'idée proposée s'appuie sur la densité du réseau ; les noeuds se trouvant dans le même endroit, captent la même information. Chaque noeud membre reçoit des données captées auprès de ses voisins. Ces données servent comme des variables d'entrée du réseau de neurones implémenté sur ce noeud membre. Ce dernier exécute ensuite le mécanisme d'apprentissage et compare le résultat avec sa donnée captée (afin de vérifier si sa donnée captée ressemble aux données reçues). S'il y'a une très grande différence, cela signifie que la donnée captée est erronée (à cause de l'environnement hostile ou un manque d'énergie). Si la donnée n'est pas valide alors on n'a pas besoin de l'envoyer au CH ce qui permet de réduire le nombre de messages envoyés et de ne pas falsifier le résultat de l'agrégation de données, et par conséquent, minimiser la consommation d'énergie. Toutefois, cette méthode nécessite un clustering statique vu que les réseaux de neurones sont déjà implémentés sur les noeuds membres.

D'autres travaux se sont concentrés sur les protocoles de routage qui adoptent le mécanisme du clustering dynamique. Les auteurs de [101, 72, 73] ont pensé à exécuter des réseaux de neurones sur les CH (sachant que le nombre de CH est plus petit que le nombre de membres). Mais comme le clustering est dynamique, ces CH changent de rôle. Dans ce cas, il faut implémenter un réseau de neurones sur chaque noeud du réseau, et seuls les noeuds élus CH exécutent leurs réseaux de neurones. L'utilisation des réseaux de neurones au niveau du CH est dans le but de classifier les données et améliorer l'agrégation de données. Par exemple, la figure 3.5 illustre un exemple d'utilisation des réseaux de neurones dans la détection des incendies de forêts. Un CH reçoit les données captées par leurs membres (température et humidité maximums, minimums et moyennes), et à l'aide de ces valeurs, il calcule le résultat de la fonction d'évaluation F . Ce résultat est comparé à une valeur qui décrit l'état normal d'une forêt. Si cette valeur est beaucoup plus petite que le résultat de la fonction F , alors il y'a un risque d'incendie dans la forêt.

Par ailleurs, les auteurs dans [1] ont conçu un moyen qui permet de détecter les *hotspots* dans le réseau. C'est cette idée qu'ont prise les chercheurs dans [23] pour proposer une nouvelle stratégie pour la hiérarchisation de l'architecture de communication d'un réseau de capteurs. Ils ont en effet axé leurs études sur les avantages à prédire, à l'aide d'un réseau de neurones, l'énergie qui pourra être consommée par un noeud capteur durant le routage. Ils présumant que les réseaux de neurones constituent un bon critère d'élection des CH. L'idée consiste à implémenter, sur chaque noeud capteur, un réseau de neurones

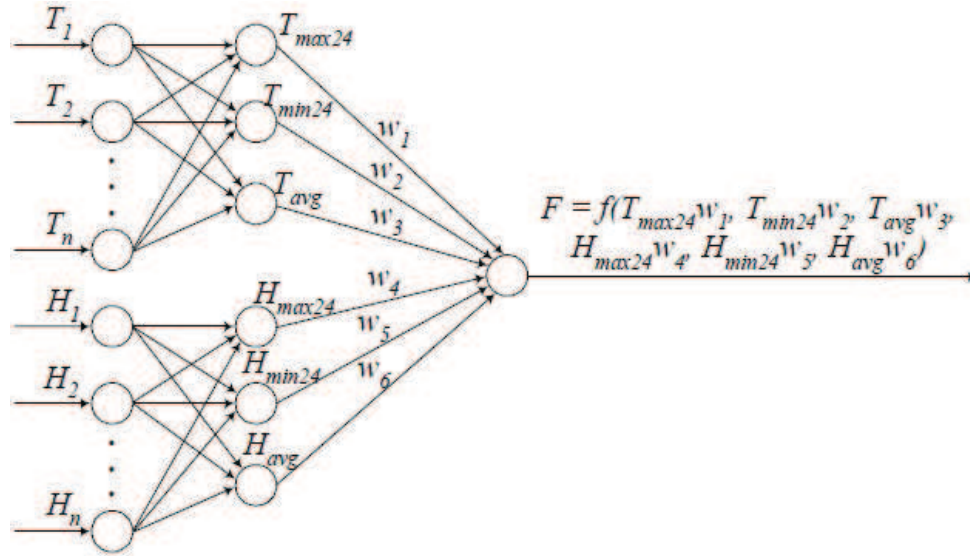


FIGURE 3.5 – Classification de données par un réseau de neurones pour la détection des incendies [101].

possédant comme entrées : la distance entre ce noeud et la station de base, le nombre de voisins et ainsi que d'autres critères. Le résultat exprime l'énergie qui pourra être consommée par ce noeud durant le routage. Les noeuds prévus pour consommer moins d'énergie, vont jouer le rôle de CH (car ils ne sont pas des *hotspots*).

D'autres travaux ont aussi visé cette classe des réseaux de capteurs ; réseaux de surveillance et de sécurité. Ils ont aussi utilisé le même principe, c'est-à-dire, ils ont implémenté des réseaux de neurones formels sur chaque noeud capteur afin de vérifier certaines propriétés tels que ; le traitement des évènements inhabituels [47], la détection du changement du signal [77] et la détection d'intrusions avec des robots mobiles intelligents [71]. D'autres chercheurs ont également étudié la notion de sécurité dans [92], et ce, en sécurisant le protocole d'accès au media MAC-CSMA (*Media Access Control-Carrier Sense Multiple Access*) contre les attaques de type dénis de service DoS (*Denial of Service*).

Pour récapituler, les réseaux de neurones formels possèdent une grande capacité d'apprentissage automatique à partir de données modélisant le problème à résoudre, ce qui permet de remplacer efficacement des modèles mathématiques pouvant être extrêmement complexes. En raison de cette propriété [15], les réseaux de neurones se sont révélés être des modèles très performants. Il faut noter que cette propriété est avantageuse, non seulement dans le cadre de la classification, mais, plus généralement, dans le cadre de l'intelligence artificielle et de l'automatique. De plus, ce sont des modèles qui conviennent pour les applications des réseaux de capteurs qui prennent en compte la fiabilité de transmission et la qualité de service QoS (*Quality of Service*). En effet, les noeuds capteurs et les liens

de communication dans les réseaux de capteurs sont souvent très vulnérables à cause de la technologie sans fil et leur déploiement à des endroits où ils ne subissent pas les conditions les plus favorables. C'est pourquoi, les réseaux de capteurs ne sont plus autant précis après quelques temps et que leurs mesures comportent des erreurs en raison du manque d'énergie, des mauvaises conditions, etc. De ce point de vue, les réseaux de neurones offrent une grande robustesse et possèdent une grande capacité de correction d'erreurs de transmission sans consommer beaucoup d'énergie ni d'espace de stockage.

Cependant, ces modèles sont de véritables boîtes noires qui ne permettent pas d'interpréter les modèles construits. Par conséquent, le manque de lisibilité des modèles créés est un frein à l'utilisation des réseaux de neurones. En effet, il est quasiment impossible de déterminer la cause en cas d'erreur du système. De plus, le problème qui se pose en général est de calculer les valeurs des coefficients synaptiques en fonction des exemples disponibles. Il faut donc trouver un mécanisme efficace qui permet de les calculer à partir des grandeurs que l'on peut acquérir du problème. Un autre défi est le fait que les réseaux de capteurs ont un problème d'échelle. En effet, un réseau de capteurs est un système complexe qui comporte un très grand nombre de noeuds. Ainsi, il faut veiller à contrôler la complexité du modèle utilisé pour un tel réseau. Il serait donc onéreux d'implémenter des réseaux de neurones sur chaque noeud du réseau, que ce soit du côté technique (intégrer un noeud de neurone sur chaque capteur) aussi bien que du côté coût de production (les noeuds deviendront plus chers ce qui a une influence sur le coût de déploiement du réseau). Par ailleurs, les réseaux de neurones sont considérés comme des outils qui visent les QoS des réseaux. Or, la plupart des réseaux de capteurs ne prennent pas en considération cette notion de QoS, par contre, ils tolèrent les fautes et les pannes des noeuds capteurs. Ainsi, les réseaux de neurones se restreignent seulement à la classe des réseaux de capteurs basés QoS et à petite échelle (un nombre réduit de noeuds capteurs).

3.3.2 Modèles à base de chaînes de Markov

Une des techniques de modélisation et d'évaluation de performances d'un système consiste à le modéliser en utilisant des probabilités. Les chaînes de Markov, qui ont été conçues en 1907 et qui prennent le nom de leur concepteur Andrey Markov [33], sont considérées comme étant l'une des méthodes basées sur les probabilités. Elles ont été utilisées depuis longtemps pour modéliser les systèmes informatiques. Elles font partie des processus stochastiques pour lesquels la théorie mathématique est bien avancée et puissante.

3.3.2.1 Présentation des chaînes de Markov

On dit qu'un processus est markovien ou sans mémoire, si son évolution future ne dépend pas du passé, mais seulement du présent (propriété de Markov) [91]. Plus précisément, une chaîne de Markov est une suite d'évènements aléatoires, dont chacun ne dépend que de l'évènement qui le précède immédiatement [7].

Une chaîne de Markov est décrite comme suit :

Soit un ensemble d'états : $S = \{s_1, s_2, \dots, s_n\}$. Le processus commence à partir de l'un de ces états et passe successivement d'un état à un autre. Ces déplacements sont appelés transitions. Si la chaîne est actuellement à l'état s_i , le déplacement vers un état suivant s_j se fait avec une probabilité p_{ij} . Le processus peut rester dans le même état s_i , et, cela se produit avec une probabilité p_{ii} . Les probabilités p_{ij} sont nommées probabilités de transitions (pour plus d'informations, voir [33]).

Sur la figure 3.6 est illustré un exemple d'une chaîne de Markov. Ce diagramme modélise une machine possédant deux états : *off* (état 0) et *on* (état 1). On passe, par exemple, de l'état 0 à l'état 1 avec une probabilité égale à α , et, on reste dans l'état 0 avec une probabilité $1 - \alpha$.

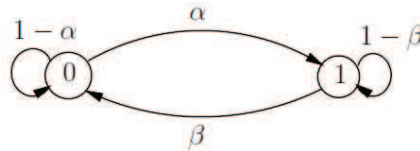


FIGURE 3.6 – Exemple d'une chaîne de Markov [85].

Après l'avoir définie, une chaîne de Markov peut être utilisée pour construire des équations analytiques qui décrivent le comportement du système et qui aident à l'évaluation de ses performances. Les domaines d'applications sont ainsi très variés : ce formalisme apparaît régulièrement en statistiques, en physique, en finances, en économie et dans bien d'autres domaines liés à l'industrie [31].

3.3.2.2 Chaînes de Markov pour les réseaux de capteurs

Les chaînes de Markov interviennent dans plusieurs applications y compris la modélisation et l'évaluation de performances dans les réseaux de capteurs. Plusieurs modèles à base de chaînes de Markov ont été invoqués tels que *Discrete-Time Markov Chain* (DTMC), *Continuous-Time Markov Chain* (CTMC), *Markov Shares*, *Dynamic Markov Random Field* (D-MRF) et *Markov Chain Monte Carlo* (MCMC).

Les chaînes de Markov prennent part à la modélisation de la notion de sécurité dans les réseaux de capteurs. Dans [52], les chercheurs ont essayé d'arriver à un bon compromis entre la minimisation de la distorsion et la minimisation de la consommation d'énergie dans un réseau de capteurs ayant une forte densité. L'auteur de l'article [44] a présenté à son tour un modèle probabiliste qui permet l'analyse des données du monde réel en tenant compte du bruit dans l'environnement, de la corruption des données et de la perte de leur intégrité. Il a démontré par la suite sa méthodologie sur un scénario précis : l'analyse des incendies des forêts, où un réseau de capteurs est mis en oeuvre pour localiser et prédire les comportements actuels et futurs des incendies. Par ailleurs, l'article [48] vise à garantir la confidentialité contre les attaques de type DoS dans les réseaux de capteurs visuels (applications de poursuite, de surveillance et de reconnaissance). Un autre domaine de sécurité a été visé par les auteurs de [71] et [10], est la détection de la présence humaine à l'aide d'un réseau de capteurs en utilisant des robots mobiles. L'idée est de calculer, à l'aide d'une chaîne de Markov, le temps durant lequel une intrusion détectée par un réseau de neurones a eu lieu. La notion de mobilité a également été traitée dans [45] où les chaînes de Markov ont été utilisées pour localiser des capteurs mobiles (en considérant non pas leurs positions exactes mais prédites) et d'effectuer un routage géographique efficace en termes d'énergie.

En plus de ces travaux qui traitent la notion de sécurité, la gestion d'énergie au niveau de la couche MAC a aussi été étudiée. Les auteurs de l'article [42] ont proposé une méthode permettant d'évaluer les performances d'un protocole MAC dans un réseau de capteurs tout en offrant la possibilité de réutiliser la spécification du protocole pour étudier diverses topologies du réseau. D'autres travaux [34, 94, 99, 21] ont évalué les performances de différents protocoles MAC afin de garantir le bon accès au média et ont modélisé les états dans lesquels peuvent se trouver les noeuds capteurs (en veille, en attente, prêt à envoyer un message, etc.). Ces états dans lesquels peut se trouver un noeud capteur ont aussi été modélisés dans [85, 86], afin de calculer, sur chaque état, l'énergie consommée par la CPU. Cet exemple est présenté dans la figure 3.7.

Le comportement d'une CPU d'un noeud capteur est représenté par un graphe dont les sommets indiquent l'activité du capteur à l'instant t . Par exemple, le sommet P_s montre que le capteur est en veille (*standby*). Le sommet P_i représente l'état repos (*idle*) d'un noeud capteur (lorsque la CPU n'effectue aucune tâche). Quant au sommet P_u , il signifie le réveil d'un noeud capteur (*power up*) dû à l'arrivée d'une tâche. Les sommets P_{o_i} montrent que le noeud capteur est en train d'exécuter la $i^{\text{ème}}$ activité. Lorsqu'un noeud capteur passe de l'activité i à l'activité j en un pas, il existera alors un arc entre deux

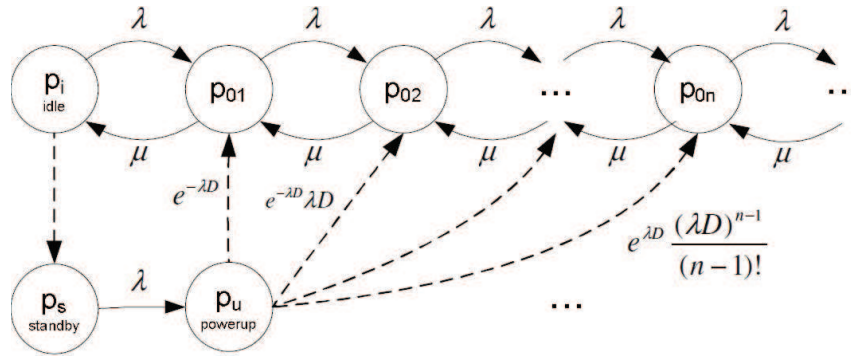


FIGURE 3.7 – Evaluer l'énergie consommée par la CPU d'un noeud capteur [85, 86].

sommets i et j du graphe. Les arcs entre les états i et j sont annotés par des probabilités de passer de l'état i à l'état j . Par exemple, l'arrivée de la première tâche permet de passer de l'état P_i à l'état P_{o_i} avec une probabilité λ . Un autre exemple est lorsqu'il n'y a plus de tâches à exécuter, alors après une certaine durée, le noeud se mettra en veille et passera à l'état P_s . A la fin, les probabilités à l'état stationnaire¹ sont calculées afin d'arriver à prédire l'énergie dissipée par la CPU.

La modélisation par chaînes de Markov est une des approches évoquées dans les réseaux de capteurs. Ces modèles sont considérés comme étant un outil très performant pour l'analyse quantitative de systèmes, et elles représentent un modèle probabiliste qui aide les noeuds capteurs à prendre des décisions indépendantes (par exemple, à l'instant t , en quel état doit se trouver un noeud capteur ; actif ou en veille) et ce en utilisant les probabilités calculées par les systèmes d'équations.

Toutefois, la modularité dont on a besoin n'est pas disponible. D'une manière simplifiée, une modification légère va entraîner la réécriture de la modélisation, et ainsi, elle induit à une nouvelle résolution d'équations (par exemple, pour un même protocole réseau et des topologies différentes, il faut réécrire toute la modélisation de ce protocole). De plus, les chaînes de Markov servent à l'évaluation de performances du système modélisé mais ne permettent pas de vérifier son comportement durant son fonctionnement (on ne peut pas analyser le système qualitativement), ce qui fait, que ce sont des modèles qui ne répondent pas à la caractéristique d'expressivité des modèles formels.

1. En raison de grand nombre d'équations, nous ne pouvons pas dérouler toutes les étapes nécessaires pour le calcul des probabilités à l'état stationnaire. Il est conseillé de se rendre aux articles [85, 86].

3.3.3 Modèles à base d'algèbres de processus

L'histoire des algèbres de processus (aussi appelées calculs de processus) remonte au début des années soixante-dix du vingtième siècle [24]. Leur apparition revient au besoin de décrire des systèmes distribués ou parallèles, que l'on nomme aussi, systèmes réactifs ou concurrents. Quand on parle des algèbres de processus, on pense immédiatement à la théorie de la concurrence. C'est pour cela que ces modèles possèdent généralement (mais pas nécessairement) une composition parallèle comme un opérateur de base.

Depuis les années 80, une multitude d'extensions et de variantes des algèbres de processus ont été proposées. Chacune modélise un paradigme de programmation particulier [24] : interactions par mémoire partagée versus communications par échange de messages, communications point à point versus multi-synchronisation (ou *broadcast*), systèmes synchrones versus systèmes asynchrones, systèmes statiques versus systèmes mobiles. D'autres extensions essaient de prendre en compte certains aspects quantitatifs tels que les propriétés probabilistes, la notion de temps, etc.

3.3.3.1 Présentation des algèbres de processus

On définit le terme algèbre de processus comme suit :

Le mot "processus" désigne le comportement d'un système. Un système ayant un comportement peut, par exemple, être une exécution d'un logiciel, des actions d'une machine ou même des actions d'un être humain. D'une manière simplifiée, un comportement est l'ensemble des évènements ou actions que peut réaliser un système, l'ordre dans lequel ils peuvent être exécutés ainsi que d'autres aspects de cette exécution telles que les probabilités et la synchronisation.

Le mot "algèbre" signifie l'utilisation d'une approche algébrique pour décrire le comportement d'un système, et ce, en utilisant les méthodes et les techniques de l'algèbre universelle (voir le livre "Algebra" de "MacLane et Birkhoff", 1967).

Une algèbre de processus est un langage défini par une syntaxe et une sémantique [30, 8] :

1. La syntaxe est à l'origine très simple. Elle détermine que les processus sont définis par une grammaire formelle simple en utilisant un nombre réduit d'opérateurs algébriques primitifs qui par assemblage, permettent de décrire des comportements complexes. Comme opérateurs de base, on utilise le symbole "||" pour la composition parallèle, le symbole ";" pour la composition séquentielle et le symbole "+" pour dénoter la composition alternative (ou le choix indéterministe).

Toutefois, les algèbres de processus ont progressivement évolué pour devenir des langages à part entière. De ce fait, leur syntaxe s'est enrichie de diverses façons afin de permettre la manipulation de données typées.

2. La sémantique permet de donner un sens aux actions des processus. Elle est définie formellement de manière axiomatique, opérationnelle ou dénotationnelle.

(a) Une sémantique axiomatique consiste en un ensemble de lois algébriques (commutativité, associativité, distributivité des opérateurs) qui permettent de démontrer l'équivalence de termes. On peut citer certaines lois de base qui sont généralement nommées lois structurelles (l'opérateur $+$ est moins prioritaire) :

- Commutativité de la composition alternative : $x + y = y + x$
- Associativité de la composition alternative : $x + (y + z) = (x + y) + z$
- Idempotence de la composition alternative : $x + x = x$
- Distributivité à droite de "+" sur ";" : $(x + y); z = x; z + y; z$
- Associativité de la composition séquentielle : $(x; y); z = x; (y; z)$
- Commutativité de la composition parallèle : $x \parallel y = y \parallel x$
- Associativité de la composition parallèle : $(x \parallel y) \parallel z = x \parallel (y \parallel z)$

On peut dire que n'importe quelle structure mathématique ayant ces trois opérateurs binaires qui satisfont ces sept lois, est une algèbre de processus.

(b) Une sémantique opérationnelle consiste en une relation de transition $E \xrightarrow{a} E'$, c'est-à-dire, un terme E peut effectuer l'action a , puis évoluer et se transformer en un terme E' . Cette relation de transition est généralement définie par induction structurelle sur la syntaxe des termes en utilisant des formats de règles standards qui garantissent par construction que la sémantique est correcte.

(c) Une sémantique dénotationnelle possède un niveau d'abstraction plus élevé que la sémantique opérationnelle ; le passage d'un état à un autre ne se fait pas selon une exécution d'une action (ou instruction élémentaire), mais, se fait selon une fonction qui transforme des entrées en sorties.

3.3.3.2 Algèbres de processus pour les réseaux de capteurs

Une large gamme d'algèbres de processus a été proposée dans la littérature dont chacune traite un aspect particulier dans les comportements des réseaux sans fil, comme le broadcast, l'asynchronisme, la mobilité, etc. (par exemple, la modélisation du protocole 802.11 [79] et la modélisation du protocole de routage AODV [64] dans les réseaux ad hoc).

En ce qui concerne les réseaux de capteurs sans fil, peu de travaux ont touché aux algèbres de processus. Les auteurs dans [49] ont proposé un nouveau langage formel intitulé PAWSN (*Process Algebra for Wireless Sensor Networks*) et un nouvel outil TEPAWSN (*Tool Environment PAWSN*). PAWSN est une algèbre de processus étendue à la notion de temps, de probabilité et notamment à la consommation d'énergie. TEPAWSN est l'environnement lié à PAWSN qui facilite la conception, l'analyse qualitative et l'analyse quantitative en traduisant les spécifications PAWSN en d'autres langages formels. Dans la même perspective, une variante de l'algèbre de processus π -calcul a été proposée dans [67] pour les réseaux de capteurs. L'algèbre π -calcul a été étendue par la notion de localité, du temps et de probabilité. L'idée de base est de considérer chaque noeud capteur comme étant un processus. Les auteurs ont démarré de la syntaxe du π -calcul donnée par :

$$P ::= \sum_{i \in I} \pi_i.P_i \mid (P_1 \mid P_2) \mid \text{new } a \text{ P} \mid !P \mid 0$$

où $\sum_{i \in I}$ représente le choix du processus P_i exécutant l'action π_i (l'opérateur $+$ représente aussi l'opérateur de choix entre les processus). Le préfixe π_i peut être :

- $x(y)$ qui signifie la réception de y à travers le canal x .
- $\bar{x}\langle y \rangle$ qui signifie l'envoi de y à travers le canal x .

$(P_1 \mid P_2)$ représente l'exécution parallèle de deux processus. $\text{new } a \text{ P}$ permet de créer un nom a et l'associer au processus P . $!P$ est utilisé pour la réplication, c'est-à-dire, plusieurs copie de P peuvent être exécutées en même temps. 0 représente la terminaison du calcul. Pour étendre ce modèle avec la notion de localité, chaque processus est muni d'un vecteur de location \vec{p} et d'une fonction de mouvement (ou déplacement) $m(\vec{p}, X)$ tel que : $m : V \times P \rightarrow V$ où X appartient à l'ensemble des processus et $V \subseteq \mathbb{R}^n$ est un espace vectoriel. Cela signifie qu'à partir de la position \vec{p} , et la fonction m , on aboutit à une nouvelle position $\vec{t} = \vec{p} + m(\vec{p}, X)$ pour laquelle se déplace un processus après une durée t . Une autre modification apportée à ce modèle est la notion de probabilité. Un opérateur de composition alternative pondérée est utilisé et défini par :

$$\sum_{i \in I} w_i P_i$$

où le symbole w_i est un poids qui représente une probabilité relative f_i au choix du processus P_i , tel que :

$$f_i = \frac{w_i}{\sum_{i \in I} w_i}$$

La nouvelle syntaxe est donnée par :

$$P_m^{\vec{p}} ::= \sum_{i \in I} \pi_i.P_{i_{m_i}}^{\vec{p}_i} \mid \sum_{j \in J} w_j.P_{j_{m_j}}^{\vec{p}_j} \mid (P_{1_{m_1}}^{\vec{p}_1} \mid P_{2_{m_2}}^{\vec{p}_2}) \mid \text{new } a \text{ P}_m^{\vec{p}} \mid !P_m^{\vec{p}} \mid 0$$

où π_i est aussi étendue et peut être :

- $x(y, r)$ qui signifie la réception de y à travers le canal x ayant un rayon r (la portée radio).
- $\bar{x}\langle y, r \rangle$ qui signifie l’envoi de y à travers le canal x ayant un rayon r .
- $wait(t)$ qui signifie le temps d’attente t .

Contrairement au π -calcul, chaque canal dans ce nouveau modèle possède un paramètre r qui sert à modéliser la portée radio d’un noeud capteur. Cela permet de limiter les interactions entre les processus voisins. Par exemple, sur la figure 3.8, P_1 et P_3 sont à portée radio alors que P_2 ne peut pas communiquer avec eux à cause de sa portée radio réduite.

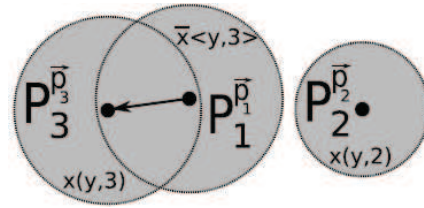


FIGURE 3.8 – Portée de communication entre les processus [67].

Cette amélioration n’est pas suffisante pour un réseau de capteurs. En effet, les communications dans un réseau de capteurs sont de type sans fil (dans l’air). Seuls les noeuds voisins peuvent communiquer. Typiquement, un processus possède plusieurs voisins, et, comme l’air est un media partagé, ce processus peut interagir simultanément avec tous ses voisins. D’où la nécessité de modéliser, en plus des communications point-à-point (entre deux processus), des communications de type un-à-plusieurs ou plusieurs-à-un. Les actions auront donc la forme suivante :

- $\overline{x_{DI(f)}}\langle y, r \rangle$ qui signifie la distribution de y à travers le canal x ayant un rayon r à l’aide d’une fonction de répartition f .
- $\overline{x_{BC}}\langle y, r \rangle$ qui signifie la diffusion de y à travers le canal x ayant un rayon r .
- $\overline{x_{AG(f,t)}}(y, r)$ qui signifie l’agrégation de y à travers le canal x ayant un rayon r à l’aide d’une fonction de répartition f et un timeout t .

Dans le but d’illustrer le fonctionnement de ce modèle dédié aux réseaux de capteurs, les auteurs ont modélisé un simple algorithme basé sur le clustering dynamique. Par exemple, devenir CH est donnée par l’équation suivante :

$$CH(id) = \psi.CH_{destr}(id) \oplus (1 - \psi).CH_{cont}(id)$$

tel que : id est l’identificateur du CH, ψ est la probabilité qu’un CH change de rôle. Si le CH change de rôle, la syntaxe est donnée par :

$$CH_{destr}(id) = \overline{cd_{BC}}\langle id \rangle.CH(id)$$

Cela veut dire qu'un CH devient membre, et pour ce faire, il diffuse son identificateur via le canal cd. Dans le cas échéant, soit il est en attente, ou bien, il répond via le canal cf, aux noeuds qui lui ont envoyés des messages.

$$CH_{cont}(id) = wait(t_{att}).CH_{att} + CH_{cf}(id)$$

Pour que les noeuds membres puissent joindre les clusters formés, le CH effectue les tâches suivantes :

$$CH_{att}(id) = \overline{cf_{BC}}\langle id, cb, q \rangle.cb_{AG(f_{ag}, t_{ag})}.CH(id)$$

Un noeud CH diffuse via le canal cf : son identificateur, ses capacités q (pour que les membres puissent joindre le CH ayant les meilleures capacités), ainsi que le canal cb à travers lequel propagent les demandes d'appartenance. Le CH reçoit ces demandes après un temps t_{att} . Les auteurs ont encore enrichi cette nouvelle algèbre de processus dans [68], en modélisant en plus des communications synchrones, des communications asynchrones. De plus, ils ont accommodé leur modèle avec la syntaxe et la sémantique de l'algèbre de processus PEPA² (*Performance Evaluation Process Algebra*) afin d'évaluer les performances du protocole modélisé. D'autres travaux ont utilisé l'algèbre PEPA dans le but d'estimer la qualité de l'information dans les réseaux de capteurs militaires [14]. On peut notamment citer les travaux de V. Galpin [29] qui modélisent et évaluent les performances du réseau avec une algèbre de processus stochastique spatiale (en étendant le modèle PEPA à la notion de location).

D'autres aspects dans les réseaux de capteurs ont été visés par les algèbres de processus comme la notion de synchronisation au niveau de la couche MAC. Le protocole FTSP (*Flooding Time Synchronization Protocol*) a été modélisé à l'aide de l'algèbre de processus CSP (*Communicating Sequential Processes*) dans [61].

Nous pouvons bien constater que les algèbres de processus présentent de nombreux avantages parmi lesquels on cite :

- la composition qui permet une construction graduelle des systèmes ;
- la substitution qui donne la possibilité de remplacer un processus par un autre ayant le même comportement, ce qui convient à l'analyse modulaire des systèmes ;
- la minimisation qui permet à un modèle basé sur une algèbre de processus d'être réduit avant l'analyse grâce aux relations d'équivalence, ce qui permet de réduire considérablement la taille des modèles.

2. L'algèbre de processus PEPA utilise des chaînes de Markov à temps continu, et elle est considérée comme étant un type des algèbres de processus stochastiques SPA (*Stochastic Process Algebra*).

On doit cependant constater qu'en dépit de ces qualités, la diffusion des algèbres de processus dans l'industrie existe, mais reste limitée. Le principal obstacle à une diffusion accrue est l'apprentissage et la maîtrise des algèbres de processus, qui exigent des efforts et du temps. Ceci peut s'expliquer par des raisons qui proviennent également de facteurs humains : les algèbres de processus constituent des modèles textuels basés sur les termes et les expressions et elles sont souvent pénalisées par leur syntaxe ésotérique, éloignée des langages de programmation usuels. Ceci est dû au fait que leurs auteurs ont fait passer les problèmes théoriques avant les questions d'ergonomie. Afin de réduire la complexité de ces modèles, plusieurs travaux de recherche se sont orientés vers la définition des passerelles permettant le passage des modèles basés sur les algèbres de processus vers des modélisations graphiques plus simples comme par exemple les automates ou les réseaux de Petri.

3.3.4 Modèles à base d'automates temporisés

Certains systèmes manipulent des aspects temps-réel de manière explicite. Ces contraintes sont de nature quantitative et concernent des durées, comme par exemple des temps de réponse. En 1990, Alur et Dill [4] ont notamment proposé le modèle des automates temporisés pour décrire le comportement de ce type de systèmes.

3.3.4.1 Présentation des automates temporisés

Les automates temporisés ont été introduits comme une extension des automates d'état finis. Leur caractéristique principale et qui leur donne cette appellation, sont les horloges qui sont des variables naturelles ou rationnelles positives, ou bien, réelles positives. Celles-ci mesurent et contrôlent la durée entre les actions du système modélisé [62, 70].

De façon générale, un automate temporisé est représenté par un graphe où chaque sommet correspond à un état du système et les arcs aux transitions instantanées entre ces états. Il possède un nombre fini d'états parmi lesquels on distingue un état dit initial et un ensemble d'états finaux. Dans chaque état, le temps peut s'écouler et cela est exprimé par la progression uniforme des valeurs d'horloges. Ainsi, dans un état, à tout instant, la valeur d'une horloge x correspond au temps écoulé depuis la dernière remise à zéro de x .

Une contrainte d'horloge est une conjonction de contraintes atomiques qui comparent la valeur d'une horloge x à une constante. Ces contraintes temporelles peuvent porter sur les états et sur les transitions [17, 16] :

- Une contrainte d’horloge qui porte sur un état est appelée invariant. C’est une propriété à vérifier lorsque le système se trouve dans cet état. Autrement dit, elle permet de restreindre le temps d’attente dans l’état, et donc, forcer l’exécution d’une transition.
- Une contrainte d’horloge qui porte sur une transition est appelée garde (*guard*). C’est une propriété à vérifier avant de pouvoir la franchir. Plus précisément, elle permet de décrire quand la transition pourra être exécutée et quelles horloges devront être remises à zéro lors du franchissement de la transition. De plus, les gardes peuvent porter des étiquettes permettant des synchronisations. Autrement dit, un système peut être représenté sous la forme d’un ensemble d’automates concurrents où la synchronisation entre eux se fait par signaux.

Un exemple système composé d’une horloge envoyant de manière périodique un signal à un utilisateur est représenté sur la figure 3.9. Les deux automates Horloge et Utilisateur communiquent par signaux.

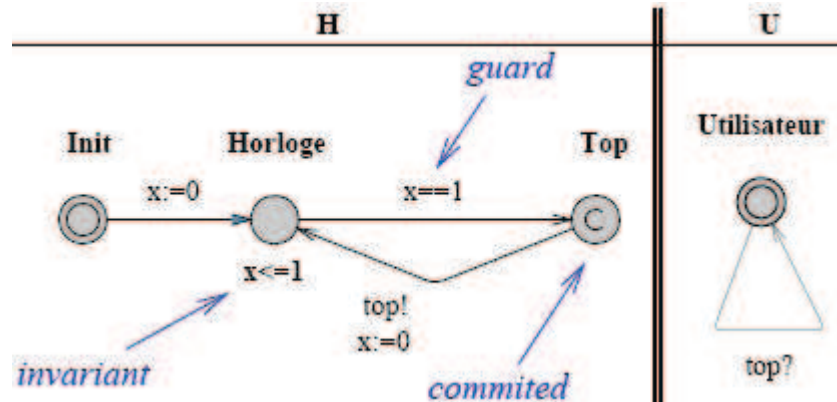


FIGURE 3.9 – Système Horloge/Utilisateur modélisé par un automate temporisé [95].

Un état peut avoir un invariant ($x \leq 1$ dans l’état central du système Horloge). L’état (*committed*) marqué C est un état où le temps ne peut pas s’écouler. Cet état doit donc être quitté immédiatement. Une transition est franchie instantanément et on peut lui affecter une garde. Le franchissement d’une transition peut également engendrer la mise à jour d’une variable (l’horloge $x := 0$), et/ou l’envoi d’un signal ($top!$). Un autre automate doit pouvoir recevoir ce signal ($top?$) sous peine de bloquer le passage de cette transition.

On peut également avoir plusieurs horloges sur le même automate temporisé. Un exemple illustratif est donné dans la figure 3.10.

On appelle trajectoire une exécution d’un automate temporisé. Une trajectoire est

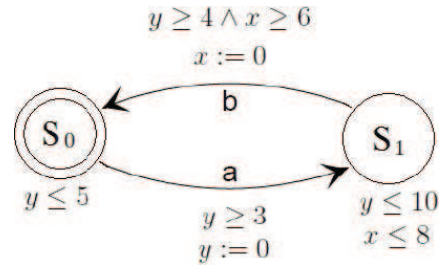


FIGURE 3.10 – Exemple d’un automate temporisé à deux horloges [95].

représentée par des n-uplets. Le premier composant d’un n-uplet représente l’état de l’automate temporisé et les autres composants représentent les horloges de cet automate. Par exemple, une trajectoire de l’automate temporisé de la figure 3.10 est une succession de triplets reliés par des flèches. Chaque flèche est étiquetée soit par une valeur réelle qui signifie l’écoulement du temps, ou bien par, l’action que doit exécuter l’automate. Plusieurs trajectoires sont possibles pour cet exemple :

1. $(S_0, 0, 0) \xrightarrow{3} (S_0, 3, 3) \xrightarrow{a} (S_1, 3, 0) \xrightarrow{4} (S_1, 7, 4) \xrightarrow{b} (S_0, 0, 4) \xrightarrow{1} (S_0, 1, 5) \xrightarrow{a} (S_1, 1, 0) \dots$
2. $(S_0, 0, 0) \xrightarrow{3.2} (S_0, 3.2, 3.2) \xrightarrow{a} (S_1, 3.2, 0) \xrightarrow{4.1} (S_1, 7.3, 4.1) \xrightarrow{b} (S_0, 0, 4.1) \xrightarrow{0.1} (S_0, 0.1, 4.2) \xrightarrow{a} (S_1, 0.1, 0) \dots$

3.3.4.2 Automates temporisés pour les réseaux de capteurs

De nos jours, de nombreuses applications de réseaux de capteurs sans fil voient le jour, dans des domaines aussi variés que la défense, la sécurité, la santé ou les maisons intelligentes. Plusieurs ont pour but de surveiller une zone géographique et de remonter une alarme en cas de détection d’un évènement redouté. Lorsqu’un tel évènement se produit, l’application doit réagir au plus tard après un temps connu et borné ; il s’agit de contraintes dites temps-réel. Dans [6], l’auteur a choisi de prendre comme exemple une application de surveillance d’accidents sur autoroute, et de proposer un nouveau protocole MAC temps-réel en faisant des hypothèses réalistes sur les réseaux de capteurs. Quant aux auteurs dans [97], ils ont choisi l’exemple des réseaux de capteurs biomédicaux (des réseaux pour les applications médicales) et ont évalué deux performances : les délais de bout en bout et le taux de transmission des paquets. Un autre travail [56] a modélisé une variante du protocole MAC, appelée LMAC (*Lightweight Medium Access Control*) dans le but de détecter et de résoudre le problème de collisions lors des transmissions de données. En effet, vu que la communication se fait par onde radio, les collisions ou bien les bruits dans le canal peuvent engendrer des erreurs dans la transmission des messages (un noeud peut détecter une collision lorsqu’il reçoit par exemple deux messages en même temps).

Dans la même direction, un protocole de synchronisation TPSN (*Time-synchronization Protocol for Sensor Networks*) a été modélisé dans [87], pour fournir une bonne synchronisation des horloges distribuées dans un réseau de capteurs à grande échelle, et ce, afin de pouvoir calculer les retards de la propagation des paquets de données dans le réseau.

On peut facilement distinguer que les automates temporisés dans les réseaux de capteurs, se focalisent beaucoup plus sur la classe de surveillance et de sécurité et modélisent surtout les contraintes de temps liées à la couche MAC. Toutefois, les automates temporisés ont été invoqués pour modéliser le routage dans la même classe de réseaux de capteurs (surveillance et sécurité). Pour la même classe de réseaux, les auteurs dans [104] et [82] ont présenté un modèle formel à l'aide des automates temporisés qui permet de calculer la durée de vie du réseau. Dans ce travail, ils se sont intéressés aux applications de détection de la présence d'un nuage radioactif, c'est-à-dire, quand les noeuds détecteront la présence de ce nuage, une alarme sera alors envoyée à la station de base.

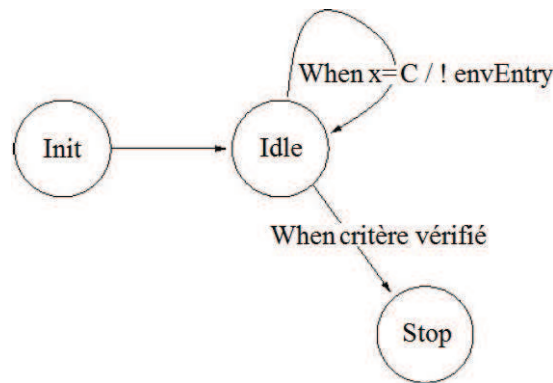


FIGURE 3.11 – Automate temporisé modélisant la détection d'un noeud radio-actif [104].

La figure 3.11 illustre l'automate lié à la détection du nuage par les noeuds capteurs. Cette détection se fait par l'envoi d'un signal *envEntry* du processus de l'environnement vers le processus du noeud (à partir de l'état *Idle*). Ce signal indique en fait la présence du nuage au-dessus des capteurs. Une fois le signal *envEntry* reçu par un noeud, un signal d'alarme est alors propagé vers la station de base. Les alarmes sont envoyées périodiquement vers le réseau (à chaque fois que l'horloge x est égale à une constante). Il est bien évident que cette application nécessite une bonne performance pendant une durée bien déterminée. En effet, l'alarme doit arriver à la station de base le plutôt possible avant que le réseau ne devienne non-fonctionnel, sinon la position de l'anomalie sera perdue. Ainsi, le temps joue un rôle très important pour l'évaluation de ce type d'applications. Dans le but d'élaborer ce modèle, deux protocoles de routage ont été modélisés pour détecter ce phénomène : le protocole *flooding* (technique d'inondation) et le protocole *Direct diffusion* afin de comparer l'impact de ces deux protocoles sur la durée de vie du

réseau. Dans cette partie, on donne juste le graphe associé à la technique d'inondation afin de comprendre l'utilisation des automates temporisés pour modéliser les protocoles de routage (voir figure 3.12). Dans cette approche, chaque noeud détectant un nuage radio-actif le diffuse à tous les noeuds voisins jusqu'à ce que le paquet arrive à la station de base.

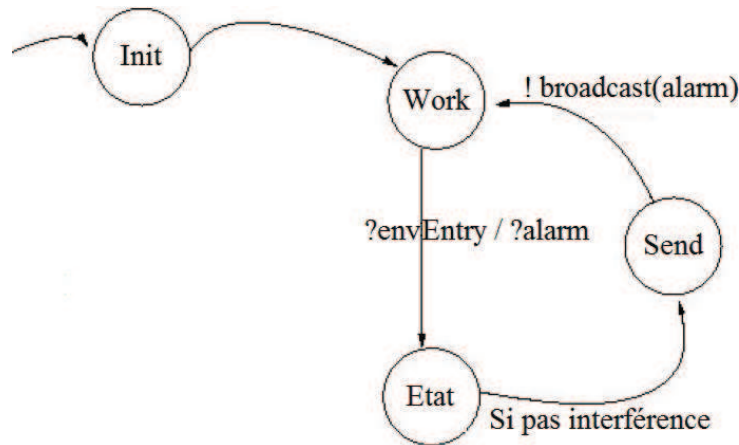


FIGURE 3.12 – Automate temporel modélisant la technique d'inondation [104].

L'automate démarre de l'état initial *Init* et les noeuds passent à l'état *Work* en attendant la réception soit des alarmes, soit des signaux de détection de nuage *envEntry*. Si c'est le cas, les noeuds passent à l'état *Etat* pour vérifier d'abord si l'un des noeuds destinataires est entrain d'utiliser sa radio. Dans le cas échéant (c'est-à-dire, s'il n'y a pas d'interférence), les noeuds passent à l'état *Send* et envoient donc le signal. Enfin, les noeuds reviennent à l'état *Work* pour attendre une nouvelle entrée.

D'après ces exemples, on peut bien conclure que les automates temporisés constituent un modèle graphique simple. Ce point est important car le fait d'écrire un modèle sous forme graphique plutôt que sous forme d'équations, permet de le rendre lisible par des personnes dont la formation scientifique n'est pas forcément poussée. De plus, ces modèles expriment aisément l'ordre des actions qui constitue une contrainte temporelle qualitative.

La limitation principale des automates temporisés est leur manque d'expressivité de la notion d'interaction. Pour modéliser deux tâches s'effectuant dans le même intervalle de temps, il faut en effet représenter par des états, toutes les dispositions possibles de juxtaposition. En généralisant, s'il y'a p processus dans le système et que chacun peut se trouver dans n états différents, le nombre de configurations possibles du système sera de l'ordre de N^p . Cette technique d'entrelacement (*interleaving*) présente une source majeure du phénomène d'explosion combinatoire. En effet, pour un grand nombre de tâches, la taille de l'automate explose.

3.3.5 Modèles à base de réseaux de Petri

Les Réseaux de Petri ou RdP (*Petri Nets*) [75] ont été conçus par Carl Adam Petri en 1962. Ils constituent un ensemble d'outils graphiques associés à des techniques mathématiques solides bien adaptées à l'expression des systèmes distribués. Etant un outil graphique, un RdP nous aide à comprendre facilement le système modélisé et permet de simuler les activités dynamiques et concurrentes.

3.3.5.1 Présentation des réseaux de Petri

Un RdP [22] est un graphe biparti comprenant deux types de sommets. Certains étant les places qui représentent les variables d'état du système et les autres étant les transitions qui représentent les actions élémentaires du système qui vont entraîner l'évolution des variables d'état du système. Ces sommets sont reliés par des arcs orientés. Un arc relie soit une place à une transition, soit une transition à une place ; jamais une place à une place ou une transition à une transition. Tout arc doit avoir à son extrémité un sommet (place ou transition). Un exemple d'un RdP est représenté sur la figure 3.13.

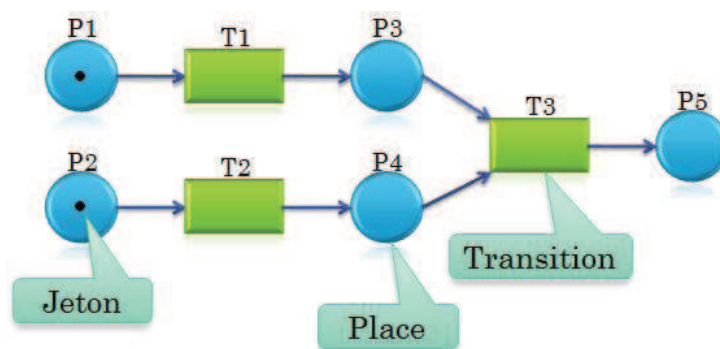


FIGURE 3.13 – Exemple d'un RdP.

A chaque place et transition est associé un nom. Par exemple, sur le RdP de la figure 3.13, les places sont nommées P_1 , P_2 , P_3 , P_4 et P_5 et les transitions T_1 , T_2 et T_3 . On dit que P_3 est en sortie ou en aval de T_1 et que T_1 est en sortie de P_1 (ou P_1 est en entrée ou en amont de T_1). A un instant donné, une place contient un certain nombre (positif ou nul) de marques ou jetons qui vont évoluer en fonction du temps : ils indiquent la valeur de la variable d'état associée à cette place à cet instant.

On appelle marquage M d'un RdP, le vecteur dont la valeur de la $i^{\text{ème}}$ composante est le nombre de marques dans la place P_i . Il indique à un instant donné l'état du RdP. Par exemple, le marquage du RdP présenté sur la figure 3.13 est donné par :

$$M_0 = [1, 1, 0, 0, 0]$$

L'évolution d'un RdP correspond au changement de son marquage au cours du temps (évolution de l'état du système) : il se traduit par un déplacement de marques. Ce déplacement aura lieu si la transition est validée. Dans ce cas, on peut effectuer le franchissement de cette transition qui consiste à :

- retirer une marque dans chacune des places en entrée de la transition ;
- ajouter une marque à chacune des places en sortie.

Par exemple, le franchissement de T_1 de la figure 3.13 mène au marquage suivant :

$$M_1 = [0, 1, 1, 0, 0]$$

Pour modéliser et analyser les applications informatiques, qui sont de plus en plus compliquées, plusieurs types de RdP ont été proposés [83]. Suivant la classification de Monika Trompedeller en 1995, trois grandes familles des RdP peuvent se distinguer selon le type du jeton :

- Les RdP caractérisés par des places contenant des valeurs booléennes. Ce sont les RdP Condition/Évènement ;
- Les RdP caractérisés par des places contenant des valeurs entières. Ce sont les RdP Etat/Transition ;
- Les RdP caractérisés par des places contenant des valeurs à niveau élevé, c'est-à-dire, des places marquées par un multi-ensemble de jetons structurés. Ce sont les RdP colorés.

Après avoir modélisé un système, on peut analyser le modèle obtenu et en déduire certaines propriétés comme l'absence du blocage, l'existence d'une solution, etc. C'est ce que l'on appelle analyse **qualitative** du système, qui permet de montrer si ce système est bon et qu'il n'engendre pas de failles lors de son fonctionnement. En plus de l'analyse qualitative, on peut avoir besoin d'une analyse dite **quantitative**, qui sert à évaluer les performances du système modélisé. Pour cela, d'autres classes de RdP ont vu le jour comme les RdP temporisés et les RdP temporels qui prennent en considération la notion du temps, ainsi que les RdP stochastiques, qui sont basés sur des processus stochastiques (comme les chaînes de Markov) et conçus pour les comportements incluant des distributions de franchissement.

3.3.5.2 Réseaux de Petri pour les réseaux de capteurs

Dû à leur clarté et simplicité, un grand nombre de formalismes basés sur les RdP ont été utilisés dans les réseaux de capteurs. En plus des différentes catégories citées dans la section précédente, d'autres types de RdP ont été utilisés pour les réseaux de capteurs comme : les RdP hiérarchiques, les RdP imbriqués et les RdP flous. La plupart ont traité l'énergie, la sécurité et la topologie du réseau.

Une approche classique concernant les réseaux sans fil en général, consiste à proposer des améliorations aux différents types de protocoles (de routage, de sécurité ou de synchronisation, etc.) et évaluer les performances des protocoles perfectionnés à l'aide des simulateurs. Actuellement, les chercheurs utilisent au lieu des simulateurs, des réseaux de Petri pour l'analyse et l'évaluation de performances des protocoles améliorés. Par exemple, pour les protocoles de synchronisation, les auteurs de [98] et [54] ont évalué les performances de certains protocoles MAC dédiés pour des applications temps-réel. Les auteurs de [27] ont conçu et validé un nouveau protocole de synchronisation basé QdS qui permet d'éviter au maximum les collisions des trames. Par ailleurs, les auteurs des articles [85] et [86] ont modélisé l'énergie consommée par la CPU à l'aide de deux modèles : les RdP stochastiques et les chaînes de Markov (exemple présenté dans de la section 3.3.2.2), et, ils les ont comparés dans le but de donner les avantages et les inconvénients des deux approches.

En ce qui concerne les protocoles de sécurité, les auteurs de [66, 81, 90, 38] ont modélisé et analysé des techniques de sécurité comme le service d'authentification et la sécurité de localisation ainsi que la gestion de clés de chiffrement afin de résister aux différentes attaques comme les DoS. D'autres applications de surveillance et de sécurité ont été analysées comme la surveillance des personnes qui se trouvent dans une mine de charbon [84], et la détection des anomalies (incendie des forêts) et la résistance aux failles [96, 41].

Par ailleurs, la modélisation de la mobilité et des changements de la topologie du réseau (à cause des failles de noeuds ou leur déplacement) a trouvé place dans [55, 51, 60, 26, 35]. De plus, dans [105, 103], des stratégies qui améliorent la connectivité des noeuds et la couverture de la zone de captage ont été prouvées. Dans [32], les auteurs ont évalué les performances d'une classe des réseaux de capteurs, intitulée réseaux de capteurs et d'actionneurs sans fil.

Concernant les protocoles de routage, certains ont été améliorés et vérifiés à l'aide des RdP comme le cas du protocole REEMR (*Reliable Energy-Efficient Multi-Level Routing*)

d'annonces (*R-Head*) et avoir envoyé son message d'appartenance au CH désiré (*S-Join*). Dans ce cas, il reçoit son slot TDMA (*R-sch*) et envoie ses données durant ce slot de temps (*S-DTH*). En dehors de son slot, un noeud membre se met en veille (*P-Sleep*) afin d'économiser son énergie. Dans le cas où un noeud non-CH ne possède pas de CH (il ne participe à aucun cluster), alors, il communiquera directement ses données à la station de base (*P-DTBS-member*) en franchissant la transition *No-CH* avec un taux $1 - \lambda$, et retourne par la suite à son état initial après la fin du round en cours.

Par ailleurs, dans [53], les auteurs ont modélisé le protocole de routage EEDMC afin de valider leur nouvelle stratégie d'élection des CH. Ils se sont basés non pas sur le comportement d'un noeud capteur, mais sur les interactions entre les noeuds capteurs au sein d'un cluster, tout en calculant l'énergie dissipée pour chaque action effectuée. Pour ce faire, un réseau de Petri coloré a été utilisé (voir figure 3.15) pour modéliser un cluster de taille 3 (un CH et deux membres). Le CH est représenté par le noeud A (à gauche), et les deux noeuds membres sont représentés par les noeuds B et C (à droite). On associe à chaque noeud capteur une place *Energy* qui contient des jetons de couleur énergie. Cette place désigne l'énergie d'un noeud capteur. A chaque franchissement d'une transition reliée à la place *Energy*, on soustrait un certain nombre de jetons de cette place, ce qui signifie la diminution de l'énergie de ce noeud.

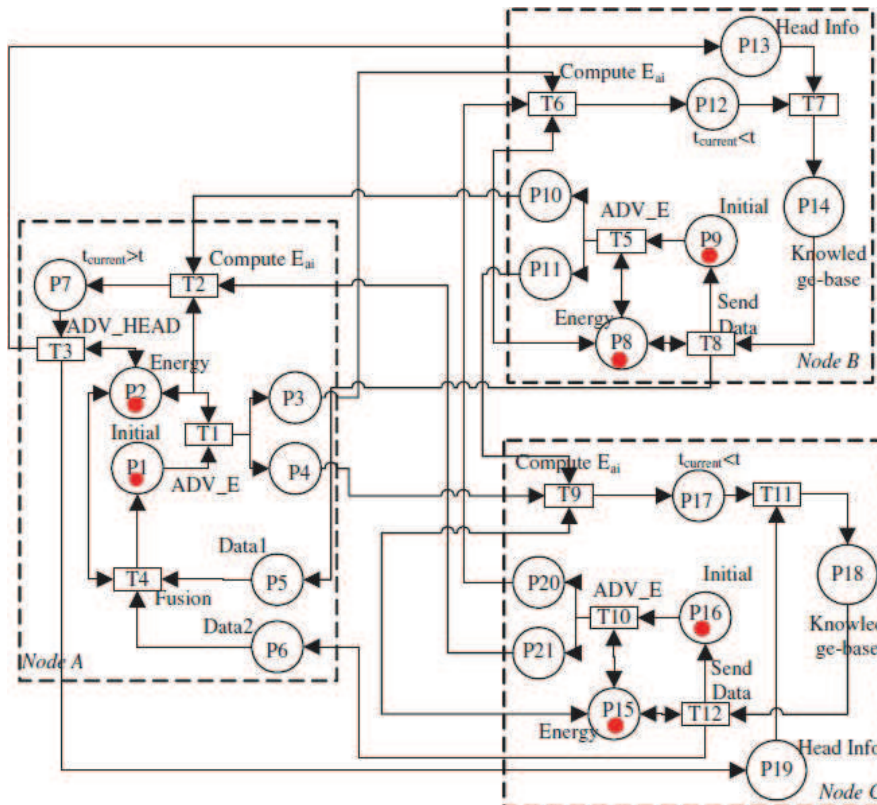


FIGURE 3.15 – Modélisation d'un algorithme d'élection des CH à l'aide d'un RdPC [53].

Au départ, chaque noeud diffuse son énergie résiduelle aux autres noeuds voisins en franchissant les transitions T_1 , T_5 et T_{10} . Ensuite, chaque noeud calcule la proportion entre son énergie résiduelle et la moyenne des énergies résiduelles de ses voisins en franchissant les transitions T_2 , T_6 , T_9 . De même, l'énergie diminue ce qui fait que le nombre de jetons dans les places *Energy* diminue. Le noeud ayant la plus grande proportion sera élu CH. Ce dernier envoie son rang aux autres noeuds en exécutant l'action T_3 et ces derniers reçoivent cette information en exécutant les transitions T_7 et T_{11} . A cet instant commence la phase de transmission. Les membres envoient leurs données captées à leur CH en tirant les transitions T_8 et T_{12} . Ce dernier agrège ses données en tirant la transition T_4 . Comme est illustré sur le modèle, le franchissement de toutes ces transitions permet de soustraire des jetons des places *Energy* de chaque noeud. Toutefois, on remarque bien que l'énergie est modélisée d'une manière discrète. Autrement dit, l'énergie de chaque noeud prend ses valeurs dans l'ensemble de entiers naturels (parce que le nombre de jetons prend des valeurs entières). Or, ce n'est pas vraiment le cas dans un réseau de capteurs parce que l'énergie évolue d'une manière continue durant le fonctionnement des noeuds capteurs.

Après avoir présenté ce que sont les RdP, et les différents modèles à base des RdP utilisés dans les réseaux de capteurs, nous allons donner un récapitulatif sur ce type de modèle. Les RdP constituent un bon support pour la modélisation des systèmes informatiques. Ceci tient aux avantages multiples qu'ils présentent. Depuis leur apparition, ils ont fait l'objet d'un nombre important de travaux de recherche, ce qui a engendré une base théorique solide et riche de techniques et d'outils d'analyse. De plus, les RdP disposent d'une représentation graphique simple et claire qui accroît la lisibilité et permet de raisonner facilement sur les systèmes modélisés. Par ailleurs, une autre motivation pour la forte utilisation des RdP revient à la représentation naturelle des caractéristiques de concurrence, de synchronisation et du parallélisme dans les systèmes modélisés.

Tout comme les automates temporisés, les réseaux de Petri souffrent de l'explosion combinatoire. L'explosion de la taille du graphe des marquages est due au nombre important de données et l'exécution concurrente de plusieurs processus. En effet, l'entrelacement est représenté par la possibilité de franchir plusieurs transitions à partir d'un même marquage, menant ainsi à des marquages différents. Ce problème qui existe déjà pour les systèmes concurrents, persiste encore plus dans le cadre de réseaux de capteurs. L'explosion de la taille du graphe des marquages est due au nombre important de noeuds capteurs dans le cas où l'on désire modéliser les interactions entre eux. Comme elle peut être due aux valeurs d'énergie modélisées comme dans le cas représenté sur la figure 3.15. En effet, l'énergie évolue au cours du temps, donc, on ne peut jamais revenir à un mar-

quage précédemment apparu. De plus, un noeud doit passer par un nombre important de valeurs d'énergie résiduelle jusqu'à ce qu'il atteigne la fin de sa durée de vie. Le nombre de marquages sera fini lorsque le réseau meurt, mais, ce nombre dans ce cas sera énorme.

3.4 Discussion et problématique

A l'issue de l'étude que nous avons menée dans cette section, nous estimons que les RdP, parviennent à remplir l'ensemble des caractéristiques fondamentales des modèles formels (décrites dans la section 3.2). Ils parviennent aussi à répondre aux besoins de notre problématique qui consiste à estimer la durée de vie du réseau durant le processus du routage.

En effet, les réseaux de neurones formels sont considérés comme étant des outils basés-QoS qui offrent une grande fiabilité et correction d'erreurs qui surviennent dans le réseau alors que la plupart des applications des réseaux de capteurs tolèrent les pannes.

En revanche, les algèbres de processus sont plus expressives ; elles ne se restreignent pas à la QoS mais elles touchent à beaucoup de notions des réseaux de capteurs comme la consommation d'énergie durant le routage. Mais reste à dire que ce sont des techniques très difficiles à manipuler parce qu'elles assurent la caractéristique d'expressivité au détriment de la simplicité alors qu'un modèle formel doit être suffisamment expressif pour décrire le comportement des systèmes tout en restant raisonnablement analysable.

Les chaînes de Markov quant à elles sont plus appropriées pour l'évaluation de performances (comme la durée de vie du réseau) des protocoles de routage. Elles ne permettent cependant pas la description du comportement du réseau et des interactions qui peuvent exister entre les noeuds capteurs. De façon simplifiée, elles ne permettent pas une analyse qualitative du protocole de routage modélisé.

Les automates temporisés et les RdP sont plus clairs en raison de leur représentation graphique qui aide à l'analyse et l'évaluation des performances des protocoles de routage modélisés. Néanmoins, la taille du modèle à base d'automates temporisés peut être extrêmement grande à cause de leur manque d'expressivité de la notion d'interaction. C'est à ce moment que les réseaux de Petri reviennent sur le devant pour représenter la possibilité que plusieurs processus évoluent simultanément au sein du même système.

Le formalisme RdP convient bien pour modéliser les communications et les interactions entre les noeuds d'un réseau de capteurs. Un RdP est donc le modèle le plus adéquat à notre travail qui consiste en la description du routage dans les réseaux de capteurs et à l'estimation de la consommation d'énergie au sein du réseau. Cela dit, dans les RdP, la consommation d'énergie n'était pas encore la préoccupation majeure et sa quantification n'a pas été modélisée d'une manière explicite. Or, il serait plus judicieux de la rendre explicite vu qu'elle est d'une importance primordiale et qu'aucun domaine d'applications ne pourrait s'en passer. Dans un RdP, quand une action survient, on ne peut pas savoir à partir du graphe combien consomme-t-elle d'énergie. Ainsi, il nous faut répondre à une question : combien d'énergie consomme une action ?

Reprenons l'exemple de la modélisation du protocole EEDMC (figure 3.15). Cet exemple a répondu à cette question grâce aux RdP colorés, et cela, en utilisant des places *Energy* qui modélisent l'énergie des noeuds capteurs. Toutefois, cette modélisation étant de nature discrète, l'énergie des noeuds prend en réalité des valeurs continues. De plus, on associe à chaque noeud du réseau une place *Energy* que l'on relie à chaque transition qui consomme de l'énergie. En conséquence, on aura un plus grand nombre de places et d'arcs dans le modèle RdP. Pour pallier à ce problème, nous tentons de proposer un nouveau modèle à base de RdP que nous appellerons EgPN (*Energy Petri Net*). Ce nouveau modèle nous permet, au niveau de la modélisation, de gérer explicitement la consommation d'énergie. Cela permet non seulement de réduire le nombre de places et d'arcs dans le modèle, mais de modéliser l'énergie de manière continue, c'est-à-dire, l'énergie ne prend pas des valeurs entières mais réelles. Le nouveau modèle EgPN sera présenté dans le prochain chapitre.

3.5 Conclusion

L'intérêt de la modélisation d'un système est de pouvoir construire une vision abstraite de ce système afin de pouvoir analyser son comportement de façon prévisionnelle. Cette analyse peut se faire en réalisant des tests ou des simulations, mais l'apport principal des modèles formels est qu'il est possible d'analyser formellement, le comportement du système modélisé et d'évaluer certains paramètres de ses performances.

Le but de ce chapitre a été donc d'étudier un certain nombre de ces modèles, qui sont utilisés dans le cadre des réseaux de capteurs sans fil. Un grand nombre de formalismes ont été proposés, parmi lesquels, on peut mentionner les réseaux de neurones, les chaînes de Markov, les algèbres de processus, les automates temporisés ainsi que les réseaux de Petri. Le choix de chaque modèle se base sur les besoins des utilisateurs, et sur les propriétés des systèmes que l'on voudrait modéliser.

Dans le cadre de notre travail, nous nous sommes intéressés à la modélisation du routage dans les réseaux de capteurs et à l'évaluation de la consommation d'énergie durant l'acheminement de données. A l'issue de l'étude que nous avons faite dans ce chapitre, nous avons choisi le formalisme RdP qui s'est avéré le plus adéquat à notre problématique. En effet, les difficultés mathématiques qui limitent l'utilisation des chaînes de Markov, les réseaux de neurones et les algèbres de processus, rendent les modèles graphiques plus attrayants. Parmi ces modèles graphiques, nous avons étudié les automates temporisés et les RdP. Ces derniers sont considérés comme étant une extension des automates, ayant deux types de noeuds : les places et les transitions. Ils sont conçus pour pallier au manque de description d'interaction dans les automates.

Les RdP ont le double avantage de fournir une analyse qualitative comme l'absence de blocage, et de vérifier des propriétés quantitatives comme la consommation d'énergie. Celle-ci constitue une contrainte primordiale dans les réseaux de capteurs et elle se pose en premier lieu par rapport aux autres métriques de routage. Pour cela, il serait intéressant d'étendre le modèle RdP à la dimension énergétique afin de capturer ce genre de contraintes, ce qui fera l'objet du prochain chapitre.

Chapitre 4

Modèle formel pour la consommation d'énergie : Energy Petri Net

4.1 Introduction

Pour les réseaux de capteurs, où l'énergie est un facteur essentiel, les modèles formels existants y compris les RdP n'ont pas le pouvoir d'expression de cet aspect énergétique. Au mieux, nous pouvons évaluer la consommation d'énergie avec des formalismes stochastiques, mais elle n'a pas été modélisée explicitement sur le modèle. Une initiative dans [53] consiste à utiliser des RdP colorés pour quantifier l'énergie consommée par une place *Energy* pour chaque noeud capteur. Malgré cela, le nombre de places du modèle augmente, ainsi que le nombre d'arcs qui relie toutes les transitions à ces places *Energy*. En outre, cette solution étant insuffisante, elle permet de manipuler des valeurs entières (cas discret) pour représenter l'énergie consommée, ce qui n'est pas vraiment le cas dans les réseaux de capteurs.

La contribution de ce travail s'insère dans ce cadre spécifique et consiste à proposer un nouveau modèle pour la modélisation et l'évaluation de la consommation d'énergie dans les réseaux de capteurs. Ce nouveau modèle, intitulé *Energy Petri Net* ou EgPN est une extension des RdP places/transitions augmentée à la dimension énergétique. Dans ce modèle, nous introduisons la notion du seuil énergétique sur toutes les transitions. Une transition ne peut être franchie que si l'énergie résiduelle d'un noeud capteur est supérieure à l'énergie seuil définie sur cette transition. Dans ce cas, l'énergie est modélisée de manière explicite et prend des valeurs réelles (cas continu). De plus, cela permet de réduire le nombre de places et d'arcs dans le modèle.

Le chapitre courant est dévolu à la description de notre modèle EgPN. Nous allons d'abord donner la définition formelle et la sémantique du modèle proposé. A la fin, nous montrons le pouvoir d'expression de ce modèle en termes d'énergie, et cela à travers deux

exemples ; la modélisation du clustering statique et la modélisation du protocole LEACH. Le but de la modélisation de ces deux protocoles de routage est d'analyser l'impact de la couche de routage sur la durée de vie d'un réseau de capteurs.

4.2 Formalisation du modèle

Nous avons donné une idée intuitive de ce que sont les EgPN. Il s'agit cependant d'un modèle formel. A ce titre, il convient d'en donner une définition rigoureuse.

Définition 1 On appelle un EgPN tout quintuplet $N=(P, T, W, e, Res)$ où :

- P est un ensemble fini de places tel que : $P = p_1, \dots, p_n$.
- T est un ensemble fini de transitions, disjoint de P , tel que : $T = t_1, \dots, t_m$.
- $W : P \times T \cup T \times P \rightarrow \mathbb{N}$ est la fonction de valuation des arcs.
- $e : T \rightarrow \mathbb{R}$ est la fonction qui associe à chaque transition une énergie seuil.
- $Res(Res_1, Res_2, \dots, Res_l)$ est le vecteur de l'énergie restante dans le réseau tel que Res_i est l'énergie résiduelle du $i^{\text{ème}}$ noeud capteur et l est le nombre total des noeuds capteurs dans le réseau.

La fonction de valuation W rassemble les préconditions et les postconditions. On peut les définir comme suit :

Pré : $T \times P \rightarrow \mathbb{N}$ est la restriction de $P \times T$. On note $Pré(t, p) = W(p, t)$.

Post : $T \times P \rightarrow \mathbb{N}$ est la restriction de $T \times P$. On note $Post(t, p) = W(t, p)$.

D'un point de vue graphique, les conventions pour les places, les transitions et les arcs dans un EgPN sont les mêmes qu'un réseau de Petri places/transitions. En ce qui concerne les transitions, elles sont pondérées par les seuils énergétiques (ils sont placés à côté des transitions entre parenthèses).

Exemple 1 La figure 4.1 illustre un exemple d'un EgPN défini comme suit :

- $P = \{Ready-to-Send, Msg-Sent\}$.
- $T = \{Send\}$.
- Tous les arcs sont étiquetés 1.
- $e(Send) = 0.5$.
- Et on a comme hypothèse le vecteur Res défini par $Res(15)$ ce qui signifie qu'il y a un seul noeud capteur dans le réseau tel que son énergie résiduelle est égale à 15 mJ (milli Joules).

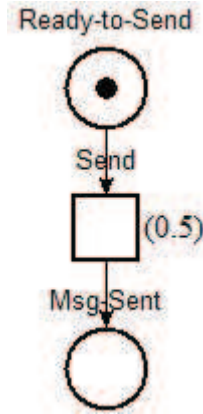


FIGURE 4.1 – Exemple d'un EgPN.

Afin de disposer d'une description statique complète du modèle, nous devons compléter la définition précédente de manière à décrire les évolutions possibles du système. Cela se fait via le marquage.

Définition 2 *Un marquage M d'un EgPN $N=(P,T,W,e,Res)$ est toute fonction $M : P \rightarrow \mathbb{N}$. On appelle un EgPN marqué tout couple (N, M_0) où M_0 est le marquage initial.*

Les marquages décrivent les évolutions possibles du système sans tenir compte des spécificités énergétiques du modèle. Nous introduirons alors la notion d'état d'un réseau EgPN qui décrit le système en prenant en considération ces spécificités. Ainsi, la description du fonctionnement d'un réseau EgPN s'appuie sur la description de l'état du réseau, état qui doit tenir compte la notion d'énergie.

Définition 3 *On appelle état d'un réseau EgPN, le couple $\xi = (M, Res)$ où :*

- *M est le marquage qui indique le nombre de jetons dans chaque place.*
- *Res est le vecteur résiduel qui indique l'énergie restante de chaque noeud capteur.*

Les définitions données ci-dessus concernent la modélisation statique. Cependant, un EgPN permet, comme tout autre réseau de Petri, de décrire l'évolution des systèmes étudiés. Il est par conséquent nécessaire de disposer d'une description de fonctionnement de ces systèmes, autrement dit, d'une étude comportementale. Les définitions suivantes permettent d'exprimer la sémantique (fonctionnement dynamique ou comportement) du modèle EgPN.

4.3 Etude comportementale - Aspects dynamiques

Dans ce qui suit, nous exprimons formellement, d'une part les règles de franchissement d'une transition et d'autre part les effets sur le système après le franchissement de cette transition. Cela se traduit par la modification du marquage et l'état du réseau (on parle alors du graphe des marquages et du graphe d'états).

4.3.1 Tir de transitions

Avant de présenter les règles de tir dans un EgPN, nous présentons d'abord ce que sont les transitions valides et les transitions tirables [22].

Définition 4 Une transition t est **valide** (ou **sensibilisée**) à partir d'un marquage M si et seulement si on a : $\forall p \in P, \text{Pré}(t, p) \leq M(p)$. Il s'agit en fait de la notion de franchissabilité dans les réseaux places/transitions. On le note $M(t \geq)$.

Définition 5 Une transition t est **tirable** (ou **franchissable**) à partir d'un marquage M si et seulement si la transition est valide et que les contraintes énergétiques sont vérifiées.

Dans un EgPN, le **tir** (ou **franchissement**) d'une transition doit prendre en considération la validité de cette transition ainsi que la vérification des conditions énergétiques. Il s'en suit qu'il nous faudra ré-énoncer la règle de tir pour les EgPN. En effet, dans les réseaux places/transitions, si toute transition valide est tirable, ce ne sera plus le cas dans les EgPN. Il faut que l'énergie résiduelle du noeud capteur soit plus grande que l'énergie seuil définie sur la transition sensibilisée.

Définition 6 Dans un EgPN, une transition t est tirable à partir d'un marquage M si et seulement si :

- $M(t \geq)$
- $e(t) \leq \text{Res}_i$ tel que Res_i est l'énergie résiduelle du $i^{\text{ème}}$ noeud capteur désirant effectuer l'action t .

Le tir de t conduit le système dans l'état décrit par le marquage M' défini par : $\forall p \in P, M'(p) = M(p) - \text{Pré}(t, p) + \text{Post}(t, p)$. On le note $M(t > M')$. De plus, sur chaque noeud i sur lequel porte l'action, $\text{Res}_{i_{\text{New}}} = \text{Res}_{i_{\text{Old}}} - e(t)$.

Dans l'exemple illustré par la figure 4.1, un noeud capteur ayant l'intention d'émettre un message, doit se mettre à l'état *Ready-to-Send*. Ce noeud ne peut effectuer cette action (émission) sauf si son énergie résiduelle est supérieure ou égale à 2.5 mJoules. Une fois la transition *Send* est franchie, la place *Msg-Sent* est marquée et l'énergie résiduelle du noeud est diminué ($\text{Res} = 15 - 0.5 = 14.5$ mJ).

4.3.2 Graphe des marquages

Le graphe des marquages d'un EgPN est un graphe orienté dont les sommets sont étiquetés par des marquages et dont les arêtes sont étiquetées par des transitions.

Définition 7 On appelle graphe des marquages d'un EgPN (N, M_0) le graphe étiqueté tel que :

- Le sommet initial est étiqueté par M_0 .
- Si un sommet ν d'étiquette M et si t est une transition tel que $M(t) > M'$, alors, il existe un sommet ν' d'étiquette M' et un arc $M \xrightarrow{t} M'$.

Exemple 2 Considérons le réseau de la figure 4.2 sur lequel, on modélise l'émission des données par deux noeuds capteurs vers un noeud collecteur.

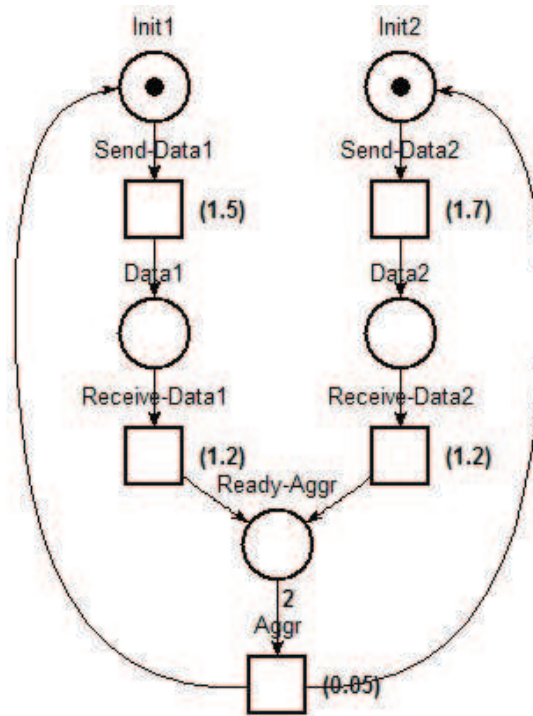


FIGURE 4.2 – Envoi et réception des données.

Supposons que le premier noeud possède une capacité de 60 mJ, le deuxième noeud 50 mJ et le troisième noeud 70 mJ. La place *Init1* correspond à l'état initial du premier noeud, quant à la place *Init2* à celui du deuxième noeud. Pour envoyer une information, les transitions *Send-Data_i* sont franchies. Dans ce cas, le premier noeud consomme 1.5 mJ et le deuxième noeud consomme 1.7 mJ pour le deuxième noeud. Le noeud collecteur reçoit les données en tirant les transitions *Receive-Data_i* en dissipant 1.2 mJ pour chaque réception. A la fin, il effectue l'opération d'agrégation de données en tirant la transition *Aggr* avec 0.05 mJ, et puis, les noeuds capteurs reviennent à l'état initial.

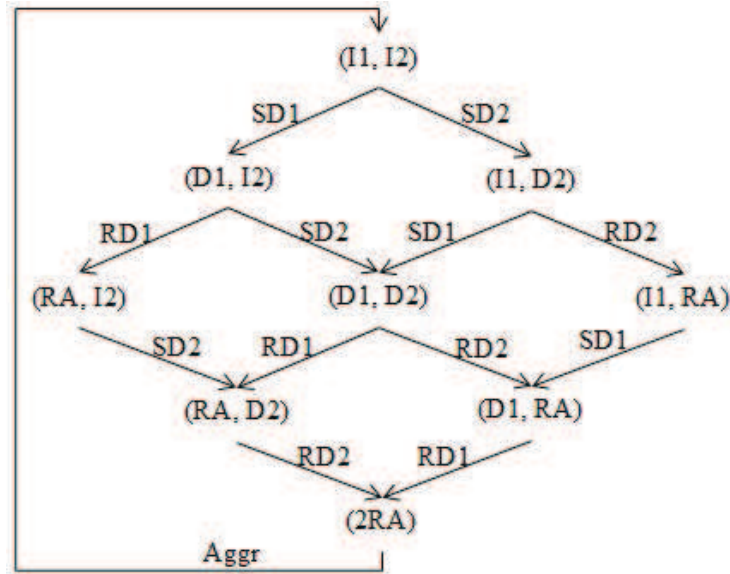


FIGURE 4.3 – Graphe des marquages associé à l'exemple de la figure 4.2.

Le graphe des marquages est donné sur la figure 4.3. Nous avons adopté une notation abrégée pour représenter les sommets du graphe : pour le premier noeud, I1 (resp. SD1, RD1, SD1) signifie *Init1* (resp. *Capt-Data1*, *Receive-Data1*, *Send-Data1*). Idem pour le deuxième noeud. Pour la place *Ready-Aggr*, on la note RA. Prenons par exemple le marquage (I1,I2) : ce dernier signifie que les places *Init1* et *Init2* contiennent chacune un jeton, les autres places étant vides. De plus, le nombre d'occurrences de chaque place correspond au nombre de jetons qu'elle contient : par exemple, (2RA) signifie que la place *Ready-Aggr* contient 2 jetons, et, les autres places sont vides.

Remarque : Nous supposons dans cet exemple que les données envoyées par les deux noeuds capteurs, arrivent à la station de base, c'est-à-dire, il n'y a pas de perte de paquets durant la transmission afin d'éviter le problème de blocage. Ce point sera détaillé plus loin.

Pour décrire la consommation d'énergie sur le graphe ci-dessus, nous introduisons des adaptations de la notion de graphe des marquages. Les noeuds sont étiquetés par des états au lieu des marquages. Nous aboutissons alors au graphe d'états.

4.3.3 Graphe d'états

Le graphe d'états d'un EgPN est un graphe orienté dont les sommets sont étiquetés par des états et dont les arêtes sont étiquetées par des transitions et des énergies seuils associées à ces transitions.

Définition 8 On appelle graphe d'états d'un EgPN (N, E_0) le graphe étiqueté tel que :

- Le sommet initial est étiqueté par E_0 tel que $E_0 = (M_0, Res)$.
- Si un sommet ω d'étiquette $E = (M, Res)$ et si t est une transition tel que $M(t) > M'$, alors, il existe un sommet ω' d'étiquette $E' = (M', Res)$ et un arc $E \xrightarrow{t, e(t)} E'$.

Pour des raisons de simplification, nous omettons du vecteur Res, les énergies résiduelles des noeuds qui ne sont pas concernés par le tir de la transition en cours. Cela permet de réduire la taille de l'état surtout lorsque le nombre de noeuds modélisés est considérable. Un état i est donc représenté par le marquage i et par la composante Res sur laquelle porte la modification (en spécifiant la position de cette composante).

Exemple 3 Considérons le même réseau de l'exemple précédent (figure 4.2).

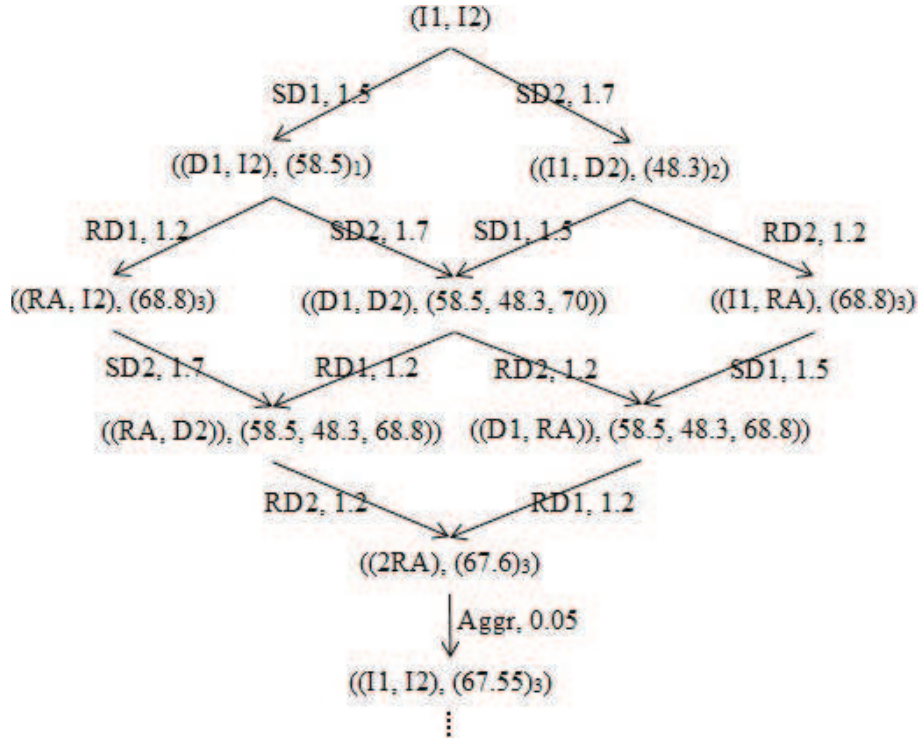


FIGURE 4.4 – Graphe d'états associé à l'exemple de la figure 4.2.

Le graphe d'états représenté par la figure 4.4 est associé au réseau représenté dans l'exemple 4.2 . L'état initial est donné $(I1, I2)$ et le vecteur résiduel est égal à :

$$\text{Res}(E_{Noeud_1}, E_{Noeud_2}, E_{Noeud_3}) = \text{Res} (60, 50, 70)$$

Deux transitions sont franchissables *Send-Data1* et *Send-Data2* car elles sont valides et $e(\text{Send-Data}_i) \leq \text{Res}_i$. Si l'on choisit par exemple de franchir la transition *Send-Data1*, l'énergie résiduelle du premier noeud diminue et on obtient $\text{Res}_1 = 60 - 1.5 = 58.5mJ$. L'état résultant est donné par : $((D1, I2), (58.5)_1)$.

L'évolution d'un état est une séquence de transitions et de prgressions en énergie. Pour explorer le graphe d'états, on suit le même processus à chaque fois que l'on rencontre une transition franchissable. Nous pouvons remarquer que le graphe d'états est très volumineux. Cela est dû à la diminution légère d'énergie par rapport à l'énergie initiale des noeuds, d'autant plus qu'elle est utilisée dans le cas continu. L'analyse de la dynamique du système modélisé devient ainsi une tâche difficile à accomplir. Pour y remédier, une solution sera proposée plus loin, notons, que cette solutions sort du cadre de ce travail et sera abordée de façon plus ample dans nos travaux futurs.

Dans les RdP en général [22], les techniques d'analyse des systèmes modélisés s'appuient sur le marquage initial. On construit un graphe des marquages pour un marquage initial donné. Tout changement de marquage initial nécessite la reprise de la construction du graphe des marquages. Il serait, pourtant, souhaitable d'analyser le système modélisé indépendamment du nombre de jetons. Une solution consiste à étudier les propriétés structurelles du réseau, c'est-à-dire, celles qui sont indépendantes du marquage initial, et qui ne dépendent que de la fonction W . Pour cela, une technique très intéressante consiste à utiliser des outils d'analyse issus de l'algèbre linéaire.

4.4 Etude structurelle - Algèbre linéaire

L'analyse d'un système modélisé à l'aide d'un RdP consiste généralement à explorer le graphe des marquages ou d'états pour un marquage ou un état initial donné. Toutefois, on doit ré-explore tout le graphe si l'on change le marquage ou l'état initial. C'est pourquoi que les chercheurs ont eu recours à une étude structurelle basée sur l'algèbre linéaire. Cette partie est dévolue à la description de l'étude structurelle de notre modèle EgPN.

4.4.1 Représentation des EgPN par matrices

La représentation matricielle des réseaux EgPN découle directement de la représentation des graphes des réseaux places/transitions. Ainsi, les matrices d'incidence, d'incidence avant et d'incidence arrière sont utilisées pour les EgPN.

Soit $N = (P, T, W, e, Res)$ un EgPN. Il est représenté à l'aide de trois matrices :

- La matrice d'incidence avant qui modélise les préconditions des transitions. Elle est définie par : $Pré = (Pré_{i,j})_{i \in 1..n, j \in 1..m}$, où $pré_{i,j} = W(p_i, t_j)$
- La matrice d'incidence arrière qui modélise les postconditions des transitions. Elle est définie par : $Post = (post_{i,j})_{i \in 1..n, j \in 1..m}$, où $post_{i,j} = W(t_j, p_i)$
- La matrice d'incidence qui modélise les effets des tirs des transitions. Elle est définie par $C = Post - Pré$.

En plus de ces matrices-là, nous devons représenter le vecteur Res . Nous le considérons comme étant un vecteur ligne à l colonnes. Enfin, le marquage initial est décrit à l'aide d'un vecteur colonne à n lignes.

Exemple 4 *Considérons le réseau de la figure 4.5 où l'on modélise un système de communication constitué de deux noeuds capteurs.*

Initialement, le premier noeud, ayant une énergie résiduelle de 20 mJ, désire envoyer une requête au deuxième noeud qui possède une énergie de 15 mJ. Une fois la transition sensibilisée *Send-Req* est franchie (car $2.5 \leq 20$), le premier noeud passe à l'état *Wait-Rep* et son énergie résiduelle passe à 17.5 mJ ($17.5 = 20 - 2.5$). La réception de la requête par le deuxième noeud requiert 2 mJ. Vu que $2 \leq 15$, cela lui permet de franchir la transition sensibilisée *Receive-Req*, ce qui conduit à la diminution de son énergie résiduelle ($15 - 2 = 13$ mJ). Une fois la requête reçue, le deuxième noeud envoie une réponse *Send-Rep* qui consomme 2.5 mJ. Ainsi, son énergie résiduelle passe à $13 - 2.5 = 10.5$ mJoules. Le deuxième noeud peut recevoir cette réponse en tirant la transition *Receive-Rep* et en consommant 2 mJ, ce qui donne une énergie résiduelle de 15.5 mJ.

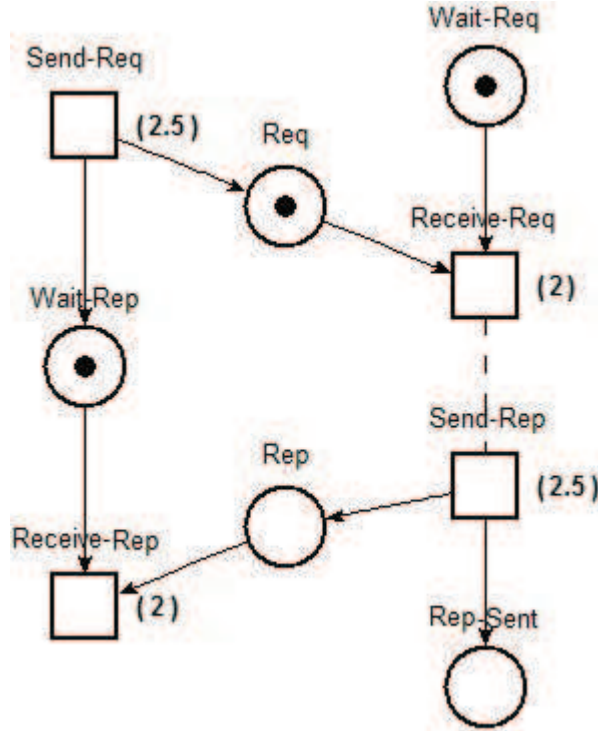


FIGURE 4.5 – Modèle d'une communication entre deux noeuds capteurs.

Nous avons :

$$\text{Pré} = \begin{matrix} & \begin{matrix} SRq & RRq & SRp & RRp \end{matrix} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} & \begin{matrix} WRq \\ Rq \\ WRp \\ Rp \\ RpS \end{matrix} \end{matrix}, \text{Post} = \begin{matrix} & \begin{matrix} SRq & RRq & SRp & RRp \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} & \begin{matrix} WRq \\ Rq \\ WRp \\ Rp \\ RpS \end{matrix} \end{matrix} \text{ et } C = \begin{matrix} & \begin{matrix} SRq & RRq & SRp & RRp \end{matrix} \\ \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix} & \begin{matrix} WRq \\ Rq \\ WRp \\ Rp \\ RpS \end{matrix} \end{matrix}$$

Le vecteur résiduel est décrit par $Res = (20, 15)$

$$\text{Le marquage initial est donné par } M_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

4.4.2 Représentation des tirs de transitions

Afin de déterminer si une transition est franchissable, on utilise la matrice $Pré$ et le vecteur Res . Pour calculer le marquage auquel on aboutit, on utilise la matrice d'incidence C et on doit mettre à jour le vecteur Res . Cependant, seule la matrice d'incidence C est nécessaire pour explorer le graphe des marquages.

Considérons la transition t_j et le marquage M . La transition t_j est franchissable depuis M si et seulement si :

1. $\forall p, \text{Pré}(t_j, p) \leq M$, c'est-à-dire, si la $j^{\text{ème}}$ colonne de Pré est inférieure ou égale à M :

$$\begin{pmatrix} W(p_1, t_j) \\ W(p_2, t_j) \\ \dots \\ W(p_n, t_j) \end{pmatrix} \leq M \text{ Soit : } \text{Pré} \times \begin{pmatrix} 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{pmatrix} \quad j \leq M$$

2. $e(t_j) \leq \text{Res}_k$ tel que Res_k est l'énergie résiduelle du noeud k désirant effectuer l'action t_j .

On aboutit alors au marquage M' défini par $M' = M - \text{Pré}(t_j, p) + \text{Post}(t_j, p), \forall p$,

$$\text{c'est-à-dire, } M' = M + C \times \begin{pmatrix} 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{pmatrix} \quad j$$

Exemple 5 Reprenons le réseau de la figure 4.5. Pour franchir la transition *Receive-Req*, nous avons :

$$M_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \forall p, \text{Pré}(RRq, p) = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \leq M_0$$

L'action étant réalisée par le deuxième noeud, nous devons vérifier si son énergie résiduelle est supérieure ou égale à l'énergie seuil de la transition *Receive-Req*, ce qui revient à vérifier si :

$$e(RRq) \leq \text{Res} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Cela donne,

$$e(RRq) = 2 \leq \begin{pmatrix} 20 & 15 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 20 \times 0 + 15 \times 1 = 15$$

Ainsi, la transition *Receive-Req* est franchissable depuis le marquage initial et son tir conduit au marquage :

$$M_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

De plus, la mise à jour du vecteur Res donne :

$$Res_{2New} = Res_{2Old} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix} - e(RRq), \text{ c'est-à-dire, } Res_{2New} = \begin{pmatrix} 20 & 15 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix} - 2 = 13$$

Dans cette partie, nous avons donné la définition formelle du modèle EgPN, ainsi que la sémantique de son fonctionnement. Pour mieux discerner le fonctionnement et l'évolution de ce modèle, nous allons dans ce qui suit, donner un exemple de modélisation afin d'évaluer la consommation d'énergie au sein du réseau. Nous nous intéressons à l'analyse de l'impact des protocoles de routage hiérarchiques sur la durée de vie du réseau.

4.5 Exemple de modélisation

Comme nous l'avons déjà vu dans le deuxième chapitre, il existe deux type de clustering : statique et dynamique. Dans cette section, nous évaluons la consommation d'énergie lorsque le réseau de capteurs adopte chacun de ces deux types. Cela revient à modéliser l'énergie consommée par les CH et l'énergie consommée par leurs membres pour chacun des deux protocoles, et, calculer par la suite l'énergie moyenne dissipée au sein du réseau. Afin d'obtenir la quantité d'énergie dépensée, il faut sommer les énergies consommées sur chacune des transitions franchies. Nous terminons cette partie par une comparaison entre la durée de vie du réseau selon le clustering statique et selon le clustering dynamique. Pour ce deuxième type, nous avons choisi le protocole LEACH parce qu'il s'agit de la source d'inspiration des autres protocoles de routage hiérarchiques.

Pour réaliser un tel travail, il faut toujours se donner des hypothèses de départ. Nous admettons les conventions considérées par le protocole LEACH¹ :

- La station de base est fixe, puissante et située loin des autres noeuds capteurs.
- Tous les noeuds sont statiques et uniformément distribués sur la zone de captage.
- Tous les noeuds sont homogènes et commencent avec la même quantité d'énergie.
- Tous les noeuds sont prévus pour capter leur environnement d'une manière continue et ils ont toujours des données à envoyer. Donc, nous nous intéressons à la classe

1. Les mêmes hypothèses ont été prises en compte par beaucoup de travaux comme dans [102] et [88].

des réseaux de collection des données d'environnement.

- Les liens de communication sont symétriques de telle sorte que l'énergie nécessaire pour transmettre un message du noeud A au noeud B est la même que l'énergie nécessaire pour transmettre un message du noeud B au noeud A .
- Une seule donnée par membre est envoyée au CH durant la phase de transmission.

En plus de ces conventions, nous supposons que :

- La durée de vie du réseau est déterminée par l'épuisement du premier noeud capteur.
- Les paquets transmis depuis leurs sources sont capables d'arriver à leurs destinations, ce qui revient à dire, qu'il n'y a pas de perte de paquets. Le besoin en cette hypothèse sera clarifié plus loin et la solution, qui permet de prendre en considération la perte des paquets, sera proposée comme perspective.

Paramètre	Valeur	Signification
$M(m)$	50	Champ de captage $M \times M$
NB	100	Nombre total des noeuds capteurs
NB_{CH}	5	Nombre de CH (5% du nombre total de noeuds capteurs)
NB_{MBR}	19	Nombre de membres dans un cluster
$d_{MBR,CH}(m)$	18	Distance moyenne entre les membres et leur CH
$d_{CH,BS}(m)$	100	Distance moyenne entre les CH et la station de base
$d_{Broadcast}(m)$	$M\sqrt{2}$	Distance nécessaire pour la diffusion
$k(bit)$	2000	Taille d'un paquet de données
$E_0(mJ)$	50	L'énergie initiale de chaque noeud capteur
$E_{elec}(nJ/bit)$	50	Energie nécessaire pour émission/réception
$E_{amp}(pJ/bit/m_2)$	100	Energie nécessaire pour amplifier le signal lors d'une émission
$E_{sense}(nJ)$	3	Energie nécessaire pour le captage de données
$E_{aggr}(nJ/bit)$	5	Energie nécessaire pour l'agrégation de données
$E_{compute}(nJ)$	5	Energie nécessaire pour le calcul des seuils T_i

TABLE 4.1 – Paramètres du réseau

Les valeurs des paramètres nécessaires pour notre modélisation sont données dans le tableau 4.1. Pour ces valeurs de paramètres, les transmissions ne sont pas à faible coût ; les protocoles de routage tentent donc de minimiser non seulement les distances de transmission, mais aussi, le nombre de communications entre les noeuds capteurs.

Concernant les distances, elles ne sont pas fixes pour le clustering dynamique. La distance entre la station de base et les noeuds capteurs est en moyenne de 100 mètres. La distance moyenne entre un CH et un membre est 18 mètres. Dans certains cas, cette distance peut dépasser ou n'atteigne pas 18 mètres. Mais 18 mètres est prise par convention de calcul alors que dans des situations pratiques, cette distance peut être plus ou moins de 18 mètres.

4.5.1 Modèle EgPN pour le clustering statique

Les rôles des noeuds capteurs étant désignés au préalable, ces noeuds ne peuvent accomplir, à la fois, toutes les tâches de captage, traitement et transmission de données. D'une manière simplifiée, les rôles des noeuds capteurs sont prédéterminés selon le clustering statique, c'est-à-dire, les CH ainsi que leurs noeuds membres sont spécifiés avant le déploiement. Après le déploiement, un membre ne peut devenir CH et un CH ne peut devenir membre dans un cluster. Les CH sont alors destinés à agréger les informations provenant de leurs noeuds membres, et envoyer le résultat final à la station de base. Cependant, les noeuds membres participent seulement au captage des informations de l'environnement et envoient ces informations à leur CH durant leur propre slot afin d'éviter le problème de collision. Donc, la réception des données par les CH ne peuvent se produire au même temps, même si cela n'est pas représenté sur le modèle. Nous nous intéressons seulement à la couche routage mais non pas à la couche MAC.

Comme les clusters sont de même taille (ils ont le même nombre de noeuds déterminé avant le déploiement) et les noeuds sont uniformément distribués, il suffit alors de modéliser l'énergie dissipée dans un seul cluster pour représenter l'énergie dissipée dans le réseau en entier. Pour réduire la complexité de notre modèle, nous prenons dans un premier lieu un exemple de cluster de taille 3 (constitué de trois noeuds capteurs). Par la suite, nous généralisons pour 20 noeuds capteurs dans un cluster. Le modèle EgPN, associé à un cluster de taille 3, est donné dans la figure 4.6. Les définitions de places et de transitions sont fournies dans les tables 4.2 et 4.3. Les énergies seuils (en milli Joules) associées aux transitions sont données dans la table 4.4.

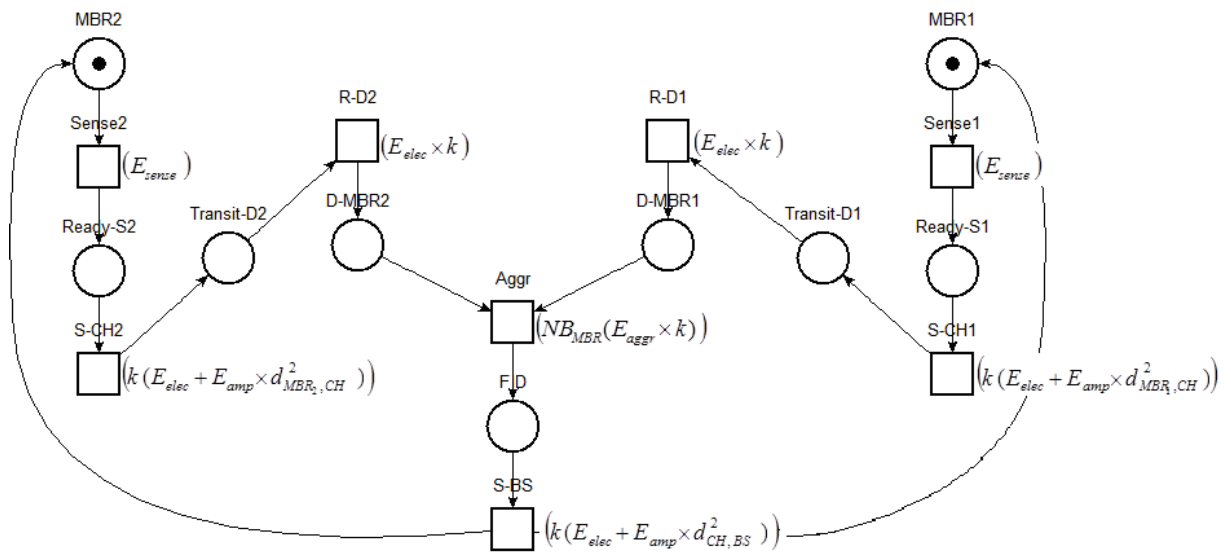


FIGURE 4.6 – Modèle EgPN pour le clustering statique.

Place	Description
MBR_i	L'état initial du $i^{\text{ème}}$ membre
$Ready-S_i$	Le $i^{\text{ème}}$ membre est prêt à envoyer le paquet de données à son CH
$Transit-D_i$	Le $i^{\text{ème}}$ membre a envoyé son paquet de données à son CH
$D-MBR_i$	Le CH a reçu le paquet de données de son $i^{\text{ème}}$ membre
$F-D$	Le paquet de données agrégées

TABLE 4.2 – Définitions des places du modèle du clustering statique.

Transition	Description
$Sense_i$	Le $i^{\text{ème}}$ membre capte les informations de son environnement
$S-CH_i$	Le $i^{\text{ème}}$ membre envoie son paquet de données à son CH
$R-D_i$	Le CH reçoit le paquet de données provenant de son $i^{\text{ème}}$ membre
$Aggr$	Le CH effectue l'opération d'agrégation de données
$S-BS$	Le CH envoie le paquet de données final à la station de base

TABLE 4.3 – Définitions des transitions du modèle du clustering statique.

Transition	$e(\text{transition})$	Valeur
$Sense_i$	E_{sense}	$3 \times 10^{-6} mJ$
$S-CH_i$	$k(E_{elec} + (E_{amp} \times d_{MBR_i,CH}^2))$	$2000(5 \times 10^{-5} + (10^{-7} \times 18^2)) = 0.1648 mJ$
$R-D_i$	$(E_{elec} \times k)$	$(5 \times 10^{-5} \times 2000) = 2 \times 10^{-1} = 0.1 mJ$
$Aggr$	$NB_{MBR}(E_{aggr} \times k)$	$2(5 \times 10^{-6} \times 2000) = 0.02 mJ$
$S-BS$	$k(E_{elec} + (E_{amp} \times d_{CH,BS}^2))$	$2000(5 \times 10^{-5} + (10^{-7} \times 100^2)) = 2.1 mJ$

TABLE 4.4 – Energies seuils des transitions du modèle du clustering statique.

Dans ce modèle, le marquage initial est donné par $M_0 = (MBR_1, MBR_2)$, et, le vecteur résiduel initial noté $Res_0 = (Res_{CH}, Res_{MBR_1}, Res_{MBR_2})$ est donné par : $Res_0 = (E_0, E_0, E_0) = (50, 50, 50)$. L'état initial résultant est donc :

$$\xi_0 = (MBR_1, MBR_2)$$

Deux évènements, qui correspondent au captage de données par les deux membres, peuvent se produire en même temps ;

- Soit on tire la transition $Sense_1$ car :

$$Pré(Sense1) \leq M_0 \text{ et } e(Sense_1) = 3 \times 10^{-6} \leq Res_{MBR_1} = 50$$

- Soit on tire la transition $Sense_2$ car :

$$Pré(Sense2) \leq M_0 \text{ et } e(Sense_2) = 3 \times 10^{-6} \leq Res_{MBR_2} = 50$$

On choisit d'exécuter la première transition, ce qui conduit à l'état :

$$\xi_1 = ((Ready-S_1, MBR_2), (50 - 3 \times 10^{-6})_2)$$

$$\xi_1 = ((Ready-S_1, MBR_2), (49.999997)_2)$$

A partir de l'état ξ_1 , on peut tirer deux transitions :

– Soit on tire la transition $S-CH_1$ car :

$$Pré(S-CH_1) \leq M_1 \text{ et } e(S-CH_1) = 0.12 \leq Res_{MBR_1} = 49.999997$$

– Soit on tire la transition $Sense_2$ car :

$$Pré(Sense_2) \leq M_1 \text{ et } e(Sense_2) = 3 \times 10^{-6} \leq Res_{MBR_2} = 50$$

On choisit d'exécuter la première transition qui correspond à l'envoi des données captées au CH à partir du premier membre. Cela conduit à l'état :

$$\xi_2 = ((Transit-D_1, MBR_2), (49.999997 - 0.1648)_2)$$

$$\xi_2 = ((Transit-D_1, MBR_2), (49.835197)_2)$$

A ce moment, deux transitions peuvent être franchies :

– Soit on tire la transition $R-D_1$ car :

$$Pré(R-D_1) \leq M_2 \text{ et } e(R-D_1) = 0.1 \leq Res_{CH} = 50$$

– Soit on tire la transition $Sense_2$ car :

$$Pré(Sense_2) \leq M_2 \text{ et } e(Sense_2) = 3 \times 10^{-6} \leq Res_{MBR_2} = 50$$

On choisit la réception de la donnée du premier membre par le CH, ce qui conduit à l'état :

$$\xi_3 = ((D-MBR_1, MBR_2), (50 - 0.1)_1)$$

$$\xi_3 = ((D-MBR_1, MBR_2), (49.9)_1)$$

Au niveau de cet état, il existe une seule transition franchissable qui correspond au captage de données par le deuxième membre. En effet,

$$Pré(Sense_2) \leq M_3 \text{ et } e(Sense_2) = 3 \times 10^{-6} \leq Res_{MBR_2} = 50$$

Son franchissement conduit à l'état :

$$\xi_4 = ((D-MBR_1, Ready-S_2), (50 - 3 \times 10^{-6})_3)$$

$$\xi_4 = ((D-MBR_1, Ready-S_2), (49.999997)_3)$$

Au niveau de l'état ξ_4 , une seule transition pourra être franchie. Elle correspond à l'envoi de données du deuxième membre au CH. En effet,

$$Pré(S-CH_2) \leq M_4 \text{ et } e(S-CH_2) = 0.12 \leq Res_{MBR_2} = 49.999997$$

Le tir de cette transition conduit à l'état :

$$\xi_5 = ((D-MBR_1, Transit-D_2), (49.999997 - 0.1648)_3)$$

$$\xi_5 = ((D-MBR_1, Transit-D_2), (49.835197)_3)$$

A ce moment, le CH est prêt à recevoir la donnée du deuxième membre. Cela se fait au moyen du tir de la transition $R-D_2$ parce que :

$$\text{Pré}(R-D_2) \leq M_5 \text{ et } e(R-D_2) = 0.1 \leq \text{Res}_{CH} = 49.9$$

Effectuer cette action conduit à l'état :

$$\xi_6 = ((D-MBR_1, D-MBR_2), (49.9 - 0.1)_1)$$

$$\xi_6 = ((D-MBR_1, D-MBR_2), (49.8)_1)$$

Une fois les données des membres reçues, le CH doit les agréger avant de les envoyer à la station de base. L'exécution de cette action se fait grâce au franchissement de la transition $Aggr$. Cela est dû à :

$$\text{Pré}(Aggr) \leq M_6 \text{ et } e(Aggr) = 0.02 \leq \text{Res}_{CH} = 49.8$$

L'état résultant est égal à :

$$\xi_7 = ((FD), (49.8 - 0.02)_1)$$

$$\xi_7 = ((FD), (49.78)_1)$$

Le paquet final est envoyé à la station de base à l'aide de la transition $S-BS$, vu que les conditions suivantes sont vérifiées :

$$\text{Pré}(S-BS) \leq M_7 \text{ et } e(S-BS) = 2.1 \leq \text{Res}_{CH} = 49.78$$

Le franchissement de cette transition nous permet d'aboutir au marquage initial. Dans ce cas, l'état du système est donné par :

$$\xi_7 = ((MBR_1, MBR_2), (49.78 - 2.1)_1)$$

$$\xi_7 = ((MBR_1, MBR_2), (47.68)_1)$$

Le même processus se répète jusqu'à l'épuisement du premier noeud, c'est-à-dire, arriver à la fin de la durée de vie du réseau. Du point de vue modélisation, on aura un blocage au niveau du système (à un moment donné, on aboutira à un état puits). En effet, après une certaine durée, une des transitions ne pourra pas être franchie parce que l'énergie résiduelle du noeud en question est inférieure à l'énergie seuil définie sur cette transition.

Dans le modèle que nous avons présenté, nous pouvons bien distinguer que les CH dissipent beaucoup plus d'énergie par rapport aux noeuds membres. En effet, un CH consomme $(50-47.68)=2.32$ mJ durant un seul round. En revanche, un noeud membre consomme $(50-49.835197)=0.164803$ mJ.

Nous pouvons aussi remarquer que l'énergie consommée par un noeud non-CH ne change pas si la taille du cluster augmente. Cependant, le CH consomme plus d'énergie si le nombre de ses membres est élevé. Cela paraît très évident puisque les CH reçoivent et agrègent plus de données captées si la taille du cluster est grande. Du point de vue modélisation, ce n'est pas commode de modéliser un cluster de 20 noeuds de la même façon qu'un cluster de taille 3. En effet, c'est déjà bien ennuyeux de modéliser un cluster de trois noeuds, alors que pour un modèle relativement grand, le nombre des marquages augmente rapidement, ce qui rend généralement son étude impossible. C'est pourquoi, il nous a fallu penser à réduire le modèle en se basant sur une technique que l'on appelle réduction [2]. Pour ce faire, nous avons eu recours à la notion de symétrie [2]. Celle-ci permet d'identifier les composants d'un système ayant des comportements similaires, c'est-à-dire, des composants exécutant les mêmes séquences d'actions dans les mêmes conditions. Etant donné que les noeuds capteurs ont le même comportement, un seul CH peut représenter tous les CH du réseau, et, un seul noeud non-CH peut représenter tous les noeuds non-CH du réseau. Cela est dû au fait que tous les CH sont homogènes. De même, les noeuds membres ont les mêmes capacités et effectuent les mêmes tâches. La figure 4.7 illustre le modèle EgPN d'un CH, et la figure 4.8 illustre celui d'un noeud membre.

Dans ce qui suit, l'énergie consommée par un CH est modélisée sur la base de l'hypothèse qu'un cluster contient 20 noeuds capteurs. La différence avec le modèle précédent est que :

$$\begin{aligned}
 e(\text{R-D}) &= NB_{MBR}(E_{elec} \times k) \\
 e(\text{R-D}) &= 19(5 \times 10^{-5} \times k) = 1.9mJ \\
 &\text{et} \\
 e(\text{Aggr}) &= NB_{MBR}(E_{aggr} \times k) \\
 e(\text{Aggr}) &= 19(5 \times 10^{-6} \times k) = 0.19mJ
 \end{aligned}$$

Ainsi, durant un round, un CH consomme :

$$\begin{aligned}
 (E_{CH})_{Round} &= e(\text{R-D}) + e(\text{Aggr}) + e(\text{S-BS}) \\
 (E_{CH})_{Round} &= 1.9 + 0.19 + 2.1 = 4.19 \text{ mJ}
 \end{aligned}$$

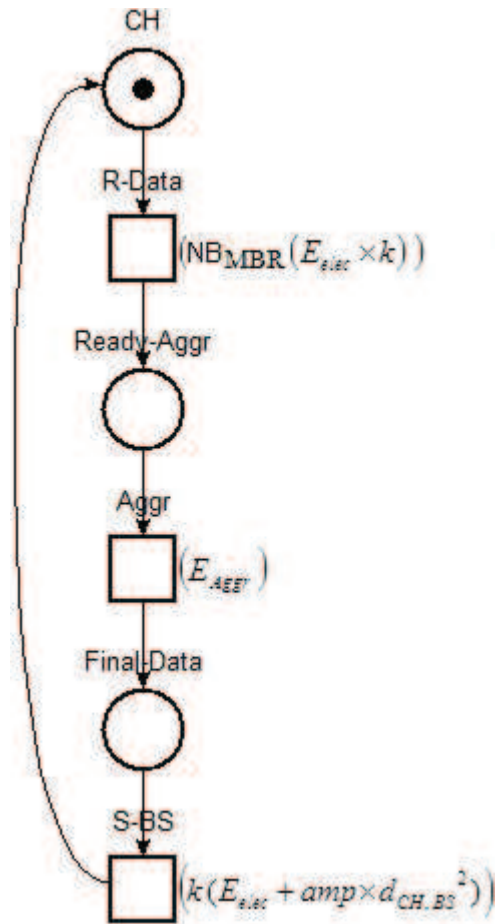


FIGURE 4.7 – Modèle EgPN d'un noeud CH selon le clustering statique.

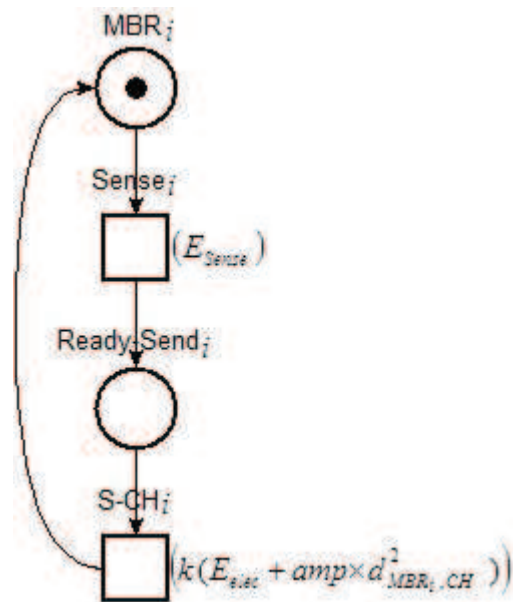


FIGURE 4.8 – Modèle EgPN d'un noeud non-CH selon le clustering statique.

Par conséquent, le taux d'énergie consommée dans un cluster durant un round est égal à :

$$(E_{Cluster})_{Round} = (E_{CH})_{Round} + 19(E_{MBR})_{Round}$$

$$(E_{Cluster})_{Round} = 4.19 + (19 \times 0.164803) \text{ mJ} = 7.321257 \text{ mJ}$$

Un CH consomme 57.23% de l'énergie totale alors qu'un membre consomme seulement 2.25%. Comme le CH ne change pas de rôle durant sa durée de vie, il sera rapidement épuisé, ce qui induit à un partitionnement du réseau et à la fin de la durée de vie du réseau au bout d'un temps très court. En effet,

- Le nombre de rounds nécessaires avant qu'un CH ne meure est égal à :
 $50/4.19=11.93 \simeq 11$ rounds.
- Le nombre de rounds nécessaires avant qu'un membre ne meure est égal à :
 $50/0.164803=303.39 \simeq 303$ rounds.

Ainsi, la durée de vie du réseau est égale à 11 rounds, et elle est vraiment trop succinte.

Une première solution consiste à déployer des CH puissants et qui n'ont pas la même énergie initiale que les noeuds non-CH afin d'avoir la même durée de vie que les noeuds non-CH. Selon cette approche, l'énergie initiale des CH doit être égale à $4.19 \times 303 \simeq 1271 \text{ mJ}$. Dans ce cas, le coût de déploiement sera plus cher, parce que les noeuds CH coûteront plus chers, sachant que le nombre total de noeuds capteurs peut atteindre des millions de noeuds, et donc, le nombre de CH dans le réseau est important.

Une deuxième solution consiste à trouver des techniques permettant d'étendre la durée de vie du réseau, non pas avec des noeuds chers, mais avec des noeuds ordinaires à prix raisonnable. Les concepteurs du protocole LEACH ont pensé à rendre le clustering dynamique de telle façon que les noeuds CH deviennent non-CH et les noeuds non-CH deviennent CH. Cela permet aux anciens membres d'effectuer les tâches les plus coûteuses en termes d'énergie afin d'alléger la charge sur les anciens CH qui sont devenus moins puissants.

4.5.2 Modèle EgPN pour le clustering dynamique

Contrairement au clustering statique, les rôles des noeuds capteurs dans le clustering dynamique ne sont pas prédéterminés. Aucun noeud capteur n'est au courant du rôle des autres noeuds. Ainsi, un noeud désirant être CH doit annoncer son rang aux autres noeuds du réseau. De plus, il doit affecter des slots de temps (TDMA) à ses membres pour que ces derniers puissent envoyer leurs données sans qu'il y'ait des collisions durant la transmission. Pour cela, les noeuds capteurs effectuent plus de tâches liées à l'ordonnancement et à l'organisation des groupes, et il y'a plus de messages en propagation (en plus des paquets de données, on a des messages de contrôle). On peut alors constater que durant un round, les noeuds consomment plus d'énergie lorsqu'ils adoptent le clustering dynamique. Cela ne signifie pas que la durée de vie du réseau sera faible. En effet, bien que les noeuds consomment plus d'énergie, le changement périodique de leurs rôles permet aux noeuds d'avoir une dissipation d'énergie équitable au sein du réseau, ce qui augmente sa durée de vie. Nous prenons comme exemple de modélisation, le protocole LEACH qui est à la base, le premier protocole de routage utilisant le mécanisme du clustering dynamique.

Comme nous l'avons déjà vu précédemment , LEACH est un protocole de routage de nature aléatoire, ce qui fait que, nous ne pouvons pas calculer le taux de la consommation d'énergie d'une façon exacte. Toutefois, nous pouvons approximativement calculer le taux d'énergie dissipée. Dans un réseau de capteurs utilisant le protocole LEACH, chaque noeud peut fonctionner d'une manière indépendante. Il peut par exemple collecter des données, envoyer des paquets, participer à l'élection des CH, etc. Pour cela, il suffit de modéliser l'énergie consommée par un seul noeud pour représenter tout le réseau (voir figure 4.9). L'ensemble des définitions des places et des transitions sont fournies dans les tables 4.5 et 4.6. Les énergies seuils (en milli Joules) associées aux transitions sont données dans la table 4.7.

Dans LEACH, le nombre de CH optimal est de 5% pour une topologie de 100 noeuds capteurs déployés sur une surface 50×50 . Donc, le pourcentage de CH désiré est égal à 0.05. Ainsi, il y'aura 5 CH (donc 5 clusters) et 95 non-CH dans le réseau. $95/5=19$ est le nombre de membres par cluster, c'est-à-dire, dans chaque cluster, il y'a au total 20 noeuds parmi lesquels, un seul est CH.

En outre, il nous faut $1/0.05=20$ rounds nécessaires pour réinitialiser l'algorithme de LEACH. En utilisant le seuil T_i de LEACH, chaque noeud devient un CH durant un seul round parmi ces 20 rounds, et reste membre durant 19 rounds. Cela signifie que, chaque 20 rounds, un noeud devient CH une seule fois et reste membre 19 fois.

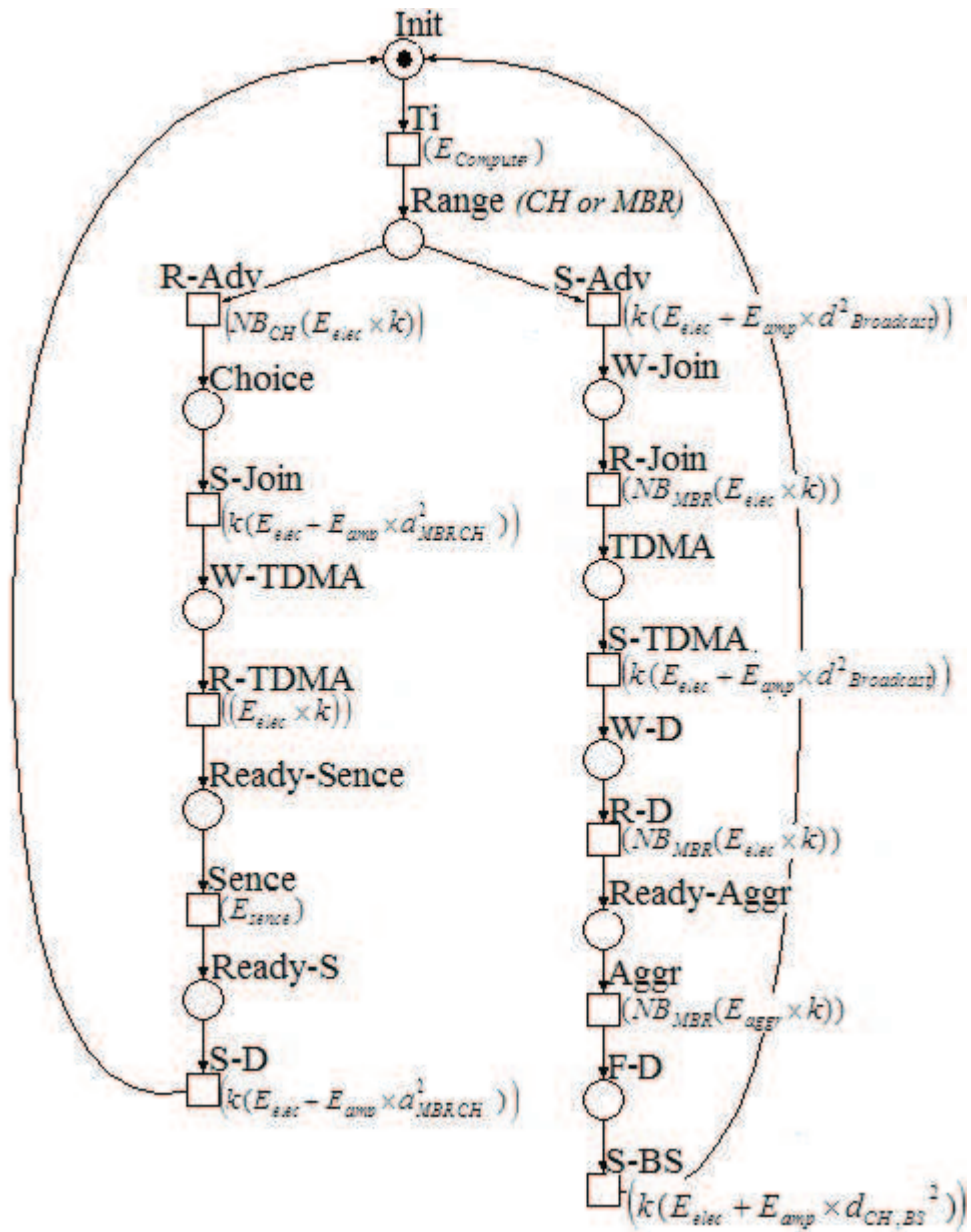


FIGURE 4.9 – Modèle EgPN pour le protocole LEACH.

Place	Description
<i>Init</i>	L'état initial du noeud capteur
<i>Range</i>	Le noeud devient soit CH soit membre
<i>W-Join</i>	Le CH attend les messages d'appartenance
<i>TDMA</i>	Le CH est prêt à diffuser le schedule TDMA
<i>W-D</i>	Le CH est prêt à recevoir les paquets de données de ses membres
<i>Ready-Aggr</i>	Le CH est prêt à agréger les paquets de données provenant de ses membres
<i>F-D</i>	Le paquet de données agrégées
<i>Choice</i>	Le noeud membre choisit un CH
<i>W-TDMA</i>	Le noeud membre attend le slot que lui est affecté
<i>Ready-Sence</i>	Le noeud membre est prêt à capter les informations de son environnement
<i>Ready-S</i>	Le noeud membre est prêt à envoyer les données à son CH

TABLE 4.5 – Définitions des places du modèle de LEACH.

Transition	Description
T_i	Calcul du seuil avec lequel un noeud capteur décide de devenir CH ou non-CH
<i>S-Adv</i>	Le CH diffuse son message d'annonce aux autres noeuds
<i>R-Join</i>	Le CH reçoit les messages d'appartenance de ses membres
<i>S-TDMA</i>	Le CH diffuse son schedule TDMA à ses noeuds membres
<i>R-D</i>	Le CH reçoit les données captées par ses membres
<i>Aggr</i>	Le CH effectue l'opération d'agrégation de données
<i>S-BS</i>	Le CH envoie le paquet de données final à la station de base
<i>R-Adv</i>	Le noeud membre reçoit les messages d'annonce des CH
<i>S-Join</i>	Le membre envoie son message d'appartenance au CH désiré
<i>R-TDMA</i>	Le membre reçoit le schedule TDMA
<i>Sence</i>	Le membre capte les données de son environnement
<i>S-D</i>	Le membre envoie les données captées à son CH

TABLE 4.6 – Définitions des transitions du modèle de LEACH.

Transition	$e(\text{transition})$	Valeur
T_i	$E_{compute}$	$5 \times 10^{-6} mJ$
<i>S-Adv</i>	$k(E_{elec} + (E_{amp} \times d_{Broadcast}^2))$	$2000(5 \times 10^{-5} + (10^{-7} \times (50\sqrt{2})^2)) = 1.1mJ$
<i>R-Join</i>	$NB_{MBR}(E_{elec} \times k)$	$19(5 \times 10^{-5} \times 2000) = 2 \times 10^{-1} = 1.9mJ$
<i>S-TDMA</i>	$k(E_{elec} + (E_{amp} \times d_{Broadcast}^2))$	$2000(5 \times 10^{-5} + (10^{-7} \times (50\sqrt{2})^2)) = 1.1mJ$
<i>R-D</i>	$NB_{MBR}(E_{elec} \times k)$	$19(5 \times 10^{-5} \times 2000) = 2 \times 10^{-1} = 1.9mJ$
<i>Aggr</i>	$NB_{MBR}(E_{aggr} \times k)$	$19(5 \times 10^{-6} \times 2000) = 0.19mJ$
<i>S-BS</i>	$k(E_{elec} + (E_{amp} \times d_{CH,BS}^2))$	$2000(5 \times 10^{-5} + (10^{-7} \times 100^2)) = 2.1mJ$
<i>R-Adv</i>	$NB_{CH}(E_{elec} \times k)$	$5(5 \times 10^{-5} \times 2000) = 2 \times 10^{-1} = 0.5mJ$
<i>S-Join</i>	$k(E_{elec} + (E_{amp} \times d_{MBR,CH}^2))$	$2000(5 \times 10^{-5} + (10^{-7} \times 18^2)) = 0.1648mJ$
<i>R-TDMA</i>	$(E_{elec} \times k)$	$(5 \times 10^{-5} \times 2000) = 2 \times 10^{-1} = 0.1mJ$
<i>Sence</i>	E_{sense}	$3 \times 10^{-6} mJ$
<i>S-D</i>	$k(E_{elec} + (E_{amp} \times d_{MBR,CH}^2))$	$2000(5 \times 10^{-5} + (10^{-7} \times 18^2)) = 0.1648mJ$

TABLE 4.7 – Energies seuils des transitions du modèle de LEACH.

Le marquage initial est donné par $M_0 = (Init)$, qui correspond à l'état initial d'un noeud capteur au début du round courant.

Le vecteur résiduel initial noté $Res_0 = (Res_{Noeud})$ est donné par : $Res_0 = (E_0) = (50)$.
L'état initial résultant est donc : $\xi_0 = ((Init), 50)$

Comme c'est illustré sur la figure 4.9, un noeud calcule d'abord le seuil T_i afin qu'il puisse décider son rang durant le round courant. Cela est réalisé par le franchissement de la transition $Compute_{T_i}$ puisque :

$$Pré(Compute_{T_i}) \leq M_0 \text{ et } e(Compute_{T_i}) = 5 \times 10^{-6} \leq Res_{Noeud} = 50$$

On aboutit à l'état suivant :

$$\xi_1 = ((Range), (50 - 5 \times 10^{-6})) \text{ d'où } \xi_1 = ((Range), (49.999995))$$

Deux cas de figures peuvent se présenter : un noeud devient CH ou reste non-CH.

- Dans le cas où le noeud devient CH, il effectuera les tâches suivantes :

Il diffuse une annonce à l'aide de la transition $S-Adv$ qui est franchissable en raison de :

$$Pré(S-Adv) \leq M_1 \text{ et } e(S-Adv) = 1.1 \leq Res_{Noeud} = 49.999995$$

L'état résultant est :

$$\xi_2 = ((W-Join), (49.999995 - 1.1)) \text{ d'où } \xi_2 = ((W-Join), (48.899995))$$

Par la suite, le CH reçoit les demandes d'appartenance. La transition $R-Join$ sera franchie puisque :

$$Pré(R-Join) \leq M_2 \text{ et } e(R-Join) = 1.9 \leq Res_{Noeud} = 48.899995$$

On aboutit à l'état suivant :

$$\xi_3 = ((TDMA), (46.999995 - 1.9)) \text{ d'où } \xi_3 = ((TDMA), (46.999995))$$

Le CH crée un *schedule TDMA* et le diffuse à ses membres. On exécute alors la transition $S-TDMA$ vu que les deux conditions suivantes sont vérifiées :

$$Pré(S-TDMA) \leq M_3 \text{ et } e(S-TDMA) = 1.1 \leq Res_{Noeud} = 46.999995$$

Le franchissement de cette transition mène vers l'état :

$$\xi_4 = ((W-D), (46.999995 - 1.1)) \text{ d'où } \xi_4 = ((W-D), (45.899995))$$

Après que le CH a affecté les slots de temps à ses membres, il reçoit à ce moment les paquets de données en exécutant l'action $R-D$. En effet,

$$Pré(R-D) \leq M_4 \text{ et } e(R-D) = 1.9 \leq Res_{Noeud} = 45.899995$$

Ainsi,

$$\xi_5 = ((Ready-Aggr), (45.899995 - 1.9)) \text{ d'où } \xi_5 = ((Ready-Aggr), (43.999995))$$

Au niveau de cet état, la transition *Aggr*, qui correspond à l'agrégation de données, est franchissable . En effet,

$$Pré(Aggr) \leq M_5 \text{ et } e(Aggr) = 0.19 \leq Res_{Noeud} = 43.999995$$

Son franchissement nous donne :

$$\xi_6 = ((F-D), (43.999995 - 0.19)) \text{ d'où } \xi_6 = ((F-D), (43.809995))$$

Au niveau de l'état ξ_6 , la transition *S-BS* pourra être franchie. Elle correspond à l'envoi du paquet de données final à la station de base. En effet,

$$Pré(S-BS) \leq M_6 \text{ et } e(S-BS) = 2.1 \leq Res_{Noeud} = 43.809995$$

Le tir de cette transition conduit à l'état :

$$\xi_7 = ((Init), (43.809995 - 2.1)) \text{ d'où } \xi_7 = ((Init), (41.709995))$$

Le noeud retourne à l'état initial. Cela signifie qu'on est arrivé à la fin du 1^{er} round où le CH a consommé en moyenne $50 - 41.709995 = 8.290005$ mJ. Le 2^{ème} round commence. Ce noeud capteur exécute encore l'action T_i pour décider son rang. Comme il a déjà été CH durant le premier round, ce noeud capteur reste non-CH durant les 19 rounds prochains. L'état du réseau est donc égal à :

$$\xi_8 = ((Range), (41.709995 - 5 \times 10^{-6})) \text{ d'où } \xi_8 = ((Range), (41.70999))$$

- Dans ce cas, le noeud est resté non-CH. Il effectuera les tâches suivantes :

Il reçoit les messages d'annonce des CH. Cela permet de franchir la transition *R-Adv* vu que :

$$Pré(R-Adv) \leq M_8 \text{ et } e(R-Adv) = 0.5 \leq Res_{Noeud} = 41.70999$$

Cela induit à :

$$\xi_9 = ((Choice), (41.70999 - 0.5)) \text{ d'où } \xi_9 = ((Choice), (41.20999))$$

Le noeud non-CH rejoint un des cinq groupes. Pour ce faire, Il doit envoyer un message d'appartenance au CH désiré en effectuant l'action *S-Join*. En effet,

$$Pré(S-Join) \leq M_9 \text{ et } e(S-Join) = 0.1648 \leq Res_{Noeud} = 41.20999$$

Le franchissement de cette transition nous permet d'aboutir à l'état :

$$\xi_{10} = ((W - TDMA), (41.70999 - 0.5)) \text{ d'où } \xi_{10} = ((Choice), (41.04519))$$

Une fois que ce noeud est devenu membre de l'un des groupes, il va recevoir son slot de temps généré par son CH. La transition *R-TDMA* sera franchie puisque :

$$\text{Pré}(R-TDMA) \leq M_{10} \text{ et } e(R-TDMA) = 0.1 \leq Res_{Noeud} = 41.04519$$

L'état résultant est égal à :

$$\xi_{11} = ((Ready-Sence), (41.04519 - 0.1)) \text{ d'où } \xi_{11} = ((Ready-Sence), (40.94519))$$

Le noeud membre est prêt à capter les informations de son environnement. L'action *Sence* est exécutée vu que :

$$\text{Pré}(Sence) \leq M_{11} \text{ et } e(Sence) = 3 \times 10^{-6} \leq Res_{Noeud} = 40.94519$$

Le noeud membre se trouve à l'état :

$$\xi_{12} = ((Ready-S), (40.94519 - 3 \times 10^{-6})) \text{ d'où } \xi_{12} = ((Ready-S), (40.945187))$$

Enfin, ce membre pourra envoyer ses données captées à son CH. Ainsi, on tire la transition *S-D* parce que :

$$\text{Pré}(S-D) \leq M_{12} \text{ et } e(S-D) = 0.1648 \leq Res_{Noeud} = 40.945187$$

Le noeud membre revient à son état initial et le 3^{ème} round est déclenché. L'état du système est donné par :

$$\xi_{13} = ((Init), (40.945187 - 0.1648)) \text{ d'où } \xi_{13} = ((Init), (40.780387))$$

Durant ce round, le noeud non-CH a consommé en moyenne $41.709995 - 40.780387 = 0.929608$ mJ.

Le même processus se répète jusqu'à l'épuisement du premier noeud, c'est-à-dire, arriver à la fin de la durée de vie du réseau.

Nous pouvons bien constater que, durant les 20 rounds, un noeud capteur consomme en moyenne $8.290005 + 19(0.929608) = 25.952557$ mJ. Le reste d'énergie $50 - 25.952557 = 24.047447$ sera consommé durant $18,53 \simeq 18$ rounds. En conséquence, chaque noeud capteur sera probablement épuisé après le 38^{ème} round (tous les noeuds meurent approximativement en même temps). Ainsi, la durée de vie du réseau est égale à environ 38 rounds.

Voici un tableau comparatif des deux techniques de routage : le clustering statique et le clustering dynamique selon le protocole LEACH (tableau 4.8). La comparaison ne porte que sur la consommation d'énergie et la durée de vie du réseau.

	Clustering Statique	LEACH
$Energie_{CH/round}$	4.19 mJ (57.23%)	8.290005 mJ (31.95%)
$Energie_{NonCH/round}$	0.164803 mJ (2.25%)	0.929608 mJ (3.58%)
<i>Durée de vie du réseau</i>	11 rounds	38 rounds

TABLE 4.8 – Consommation d'énergie dans les deux modes de clustering.

Les CH dans le cas statique, consomment 4.19 mJ durant un seul round, et donc, leur dissipation représente une proportion de 57.23% de l'énergie dissipée dans le réseau, alors qu'un noeud non-CH consomme seulement 0.164803 mJ (2.25% du total d'énergie). Tout de même, un noeud non-CH dissipe 0.929608 mJ dans le cas dynamique avec un taux de 3.58%. Ce qui revient à dire que la proportion de la consommation d'énergie d'un noeud non-CH dans le cas dynamique, est plus élevée que la proportion de sa consommation d'énergie dans le cas statique à cause de sa participation dans l'élection des CH et dans l'organisation des groupes. C'est pourquoi que le taux de dissipation d'énergie des CH dans le cas dynamique (31.95%) diminue par rapport au cas statique en dépit de leur grande consommation qui peut atteindre 8.290005 mJ.

Ce tableau illustre bien l'amélioration apportée au clustering statique par le protocole LEACH. Bien que les noeuds capteurs consomment plus d'énergie dans le cas dynamique, la durée de vie du réseau est allongée dans ce type de routage grâce à la rotation périodique des rôles des CH qui ne sont plus pénalisés par effectuer seulement des tâches gourmandes en ressources énergétiques.

4.6 Conclusion

Le travail qui a été mené dans ce chapitre, a été guidé par la proposition d'un modèle formel qui permet la modélisation et l'analyse de la consommation d'énergie dans les réseaux de capteurs sans fil. Plus précisément, ce modèle part du principe que la contrainte énergétique constitue un facteur prépondérant, notamment sur la couche réseau où les transmissions s'avèrent les plus gourmandes en termes d'énergie. Dans cette optique, nous avons proposé et décrit une nouvelle extension du modèle RdP, augmentée à la dimension énergétique, et ce, en associant des énergies seuils aux transitions. Le modèle résultant, que l'on appelle EgPN (*Energy Petri Net*), permet de rendre explicite la contrainte d'énergie.

Du point de vue modélisation, l'introduction de ce facteur nous permet d'avoir une vision claire sur la dissipation d'énergie dans le réseau. Dans le but de mieux discerner le comportement de notre modèle, nous avons abouti à la modélisation de deux techniques du clustering : statique et dynamique à l'aide du protocole LEACH. A la fin, nous avons constaté que les calculs effectués sur la consommation d'énergie pour les deux techniques, ont montré que le protocole LEACH répond bien aux critères de performances souhaités et permet de maintenir la longévité du réseau.

Du point de vue analyse, nous avons pu distinguer que le graphe d'états est très volumineux. Ainsi, l'analyse du modèle, qui dépasse le cadre de ce mémoire, devient difficile, voire impossible. Une solution sera proposée comme perspective de ce travail et sera étudiée dans des travaux futurs.

Conclusion générale

Actuellement, les réseaux de capteurs sans fil constituent un thème de recherche très dynamique et vont sans doute dans les années à venir, constituer un développement technologique majeur qui ouvre la porte à une multitude d'applications liées à la sécurité, la santé, l'agriculture, la domotique, etc. Cette large gamme d'applications accroît la complexité des réseaux de capteurs et engendre beaucoup de défis pour la conception de tels réseaux. L'énergie, étant la contrainte la plus essentielle, occupe une place importante dans les réseaux de capteurs, une place qui se consolide de plus en plus, étant donné que les noeuds capteurs fonctionnent généralement dans des environnements inaccessibles à l'aide des batteries qui ne peuvent être remplacées ou rechargées. En effet, c'est là le point le plus crucial, alors que la durée des batteries est de l'ordre de plusieurs heures en réseaux classiques, les réseaux de capteurs visent des autonomies bien plus élevées, de l'ordre de plusieurs années. Il est donc très important que les batteries durent aussi longtemps que possible et que l'utilisation des noeuds capteurs doit être efficace afin de réduire au maximum le gaspillage d'énergie. Cette efficacité doit être observée sur toutes les couches de la pile protocolaire, notamment sur la couche réseau du moment que le coût des transmissions de données, énergiquement parlant, peut atteindre un taux de 70% par rapport au coût général du fonctionnement d'un noeud capteur. L'objectif principal de cette couche est de trouver des moyens pour une mise en oeuvre des protocoles de routage économes de telle sorte que la durée de vie du réseau soit maximisée.

Les travaux de recherche qui ont été menés ont pour objectif de cerner les besoins de ces protocoles en termes de performances, particulièrement en consommation d'énergie, en s'appuyant sur des analyses de scénarios précis et réalistes. En effet, beaucoup de protocoles de routage fonctionnent très bien selon les simulateurs, mais pas aussi bien dans des applications réelles qui exigent une haute fiabilité et une exactitude élevée dans les résultats. De telles conditions nécessitent l'introduction des méthodes basées sur des principes mathématiques solides. Ces méthodes dites formelles, peuvent être utilisées comme des moyens méthodologiques pour développer de tels protocoles de manière efficace et pour assurer leur sûreté de fonctionnement.

Dans ce mémoire, nous avons exhibé les principales caractéristiques que doivent supporter les modèles formels et nous avons étudié certains de ces formalismes qui sont utilisés dans les réseaux de capteurs. Sur la base de ces caractéristiques et les objectifs que l'on souhaiterait atteindre, nous avons pu déduire que les modèles à base de RdP constituent un bon support pour la modélisation et l'analyse de différents concepts dans les réseaux de capteurs. En effet, les réseaux de neurones formels sont beaucoup plus utilisés pour les protocoles basés-QoS afin d'améliorer la fiabilité de transmission. Par ailleurs, les algèbres de processus ne se restreignent pas à la QoS mais elles touchent à beaucoup de notions des réseaux de capteurs comme la consommation d'énergie durant le routage. Toutefois, ce sont des techniques qui se basent sur les termes et les expressions, ce qui les rend très difficiles à manipuler. Quant aux chaînes de Markov, elles constituent un bon support pour l'évaluation de performances des protocoles de routage comme la durée de vie du réseau. Elles ne permettent cependant pas la description des interactions et des communications qui peuvent exister entre les noeuds capteurs. En ce qui concerne les automates temporisés, ce sont des outils graphiques simples et peuvent décrire ces interactions. En revanche, les RdP étant aussi graphiques et simples, ils sont plus expressifs que les automates temporisés.

L'étude que nous avons faite nous a permis de conclure que la modélisation formelle dans les réseaux de capteurs, requiert des compétences et encore beaucoup de recherches en sus des modèles formels déjà développés. Partant du constat que la contrainte énergétique représente une métrique de routage qui se pose en premier lieu, il serait intéressant de la modéliser explicitement sur le modèle. Nous avons alors pensé qu'il est possible de construire un modèle à base de RdP où les informations concernant la consommation d'énergie se situent sur les transitions, c'est-à-dire, les transitions sont étiquetées par des énergies. Pour parvenir à ce but, nous avons proposé une nouvelle variante des RdP, intitulée EgPN (*Energy Petri Net*), dans laquelle a été introduite la notion d'énergie d'une manière explicite. Nous avons donné une formalisation de ce modèle, son étude comportementale, ainsi que son étude structurelle. Du point de vue modélisation, l'apport de l'introduction de la dimension énergétique au modèle RdP apparaît clairement à travers le modèle EgPN. En effet, un noeud capteur ne peut effectuer une opération que dans le cas où il possède assez de capacité pour le faire. Pour illustrer le comportement d'un EgPN, nous avons modélisé et évalué l'énergie dissipée de l'une des solutions communément proposées pour la conservation d'énergie dans les réseaux de capteurs. Cette solution consiste à utiliser une architecture hiérarchisée appelée encore *clustering*. Cela nous a permis de capturer le fonctionnement des noeuds capteurs où l'énergie apparaît comme un paramètre quantifiable.

Pour conclure, nous avons jusqu'ici, expliqué la contribution visée par notre travail. Cette contribution a le double avantage d'enrichir le modèle RdP par une dimension d'énergie, et, de vérifier l'impact des techniques de routage sur la consommation d'énergie dans le réseau à l'aide du modèle résultant. D'une manière générale, le modèle EgPN que nous avons présenté dans ce travail, offre beaucoup de privilèges que peuvent offrir les RdP. En outre, ce modèle constitue une aide précieuse lors de la phase de conception d'autant plus qu'il permet de décrire d'une façon explicite la consommation d'énergie dans un réseau de capteurs. Ceci dit, notre modèle EgPN génère, comme tout autre modèle formel, un certain nombre d'insuffisances que nous allons discuter dans la prochaine section.

Perspectives et travaux futurs

Le travail que nous avons présenté dans ce mémoire comporte plusieurs pistes de recherche à explorer. Cette partie s'attache à dévoiler les différentes limitations et lacunes qu'on peut tirer du modèle EgPN et d'en proposer des solutions qui seront abordées de façon plus étendue dans nos travaux futurs.

La première serait d'analyser le comportement dynamique du modèle que nous avons présenté dans ce mémoire. Généralement, deux techniques sont envisageables : la simulation et l'analyse énumérative. Comme nous l'avons déjà vu dans le deuxième chapitre, la simulation ne permet pas de décrire toutes les possibilités d'exécution du système modélisé, et de prouver si ce dernier possède l'ensemble des propriétés souhaitées, pourtant, elle est considérée comme étant une bonne technique en raison de sa simplicité et son pouvoir de faire des mesures de performances. A propos de l'analyse énumérative, il s'agit d'explorer l'espace d'états et à les organiser en un graphe d'états qui renferme toutes les informations contenant le comportement du système modélisé. Dans le cas des EgPN, le graphe d'états est généralement très volumineux. Notons qu'avant même d'introduire la notion d'énergie sur les transitions, le graphe de marquages ou d'états des RdP qui ont été utilisés pour les réseaux de capteurs est généralement infini à cause du problème d'échelle du réseau et du nombre élevé de jetons de couleur *Energy* dans le cas des RdP colorés. Pour cela, la technique de simulation s'avère plus adéquate pour l'analyse d'un EgPN. L'élaboration d'un outil de simulation pour ce nouveau modèle permettra de visualiser le comportement des réseaux de capteurs et offrira la possibilité d'estimer la consommation d'énergie ainsi que sa durée de vie.

Une autre perspective pour ce travail serait d'introduire l'aspect temps dans le modèle EgPN qui n'exprime aucune notion du temps. En effet, bien que le temps soit souvent moins critique que l'énergie dans les réseaux de capteurs, on a également besoin de cette quantification du temps qui peut être étendue dans plusieurs directions. D'une part, il peut être confortable d'avoir une notion de temps dans le but d'estimer la durée de vie du réseau de manière explicite sur le modèle. D'autre part, il faut s'assurer que les exigences temporelles de l'application sont respectées dans les protocoles de routage. Par exemple,

dans le protocole LEACH, l'algorithme de routage est divisé en intervalles de temps associés aux différentes phases d'organisation et de transmission de telle sorte qu'aucune phase ne peut commencer qu'après l'écoulement du temps qui lui est associé. Nous pouvons également utiliser le modèle EgPN étendu au temps pour d'autres concepts importants dans les réseaux de capteurs, comme la synchronisation et l'accès au media au niveau de la couche MAC.

Notre modèle EgPN explore un autre axe de recherche qui permet de contourner le problème de pertes des paquets. En effet, la métrique qui a été employée est l'énergie consommée par bit transmis avec succès. Mais pour être réaliste, nous devons quand même tenir compte de l'importance des autres caractéristiques du réseau comme la tolérance aux fautes. Dans notre modèle, on s'est contenté de calculer seulement l'énergie dissipée lors de la transmission du paquet en question sans examiner ce qui pourrait réellement se passer (c'est-à-dire, la possibilité de le perdre avant qu'il n'atteigne sa destination). Alors, quand survient une perte des paquets, rien, dans un réseau EgPN, ne permet de la détecter, et le modèle se bloquera lorsque le CH reste en attente de ces paquets avant qu'il ne passe à l'agrégation des données. Nous pouvons donc proposer une alternative à ce problème qui consiste à quantifier les places du modèle par le temps, et, après l'écoulement de ce temps, les transitions reliées aux places vides (qui ne contiennent pas de paquets dû à leur perte) doivent être franchies afin d'éviter le blocage et de maintenir les fonctionnalités du réseau. Qui dit places vides nous fait penser aux RdP à arcs inhibiteurs.

Par ailleurs, du point de vue applicatif, nous nous sommes intéressés dans notre travail, aux réseaux de collecte des données, tandis que les réseaux de capteurs pourraient avoir de très diverses applications. Pour les versions futures de notre modèle, nous pourrions le rendre plus approprié pour les réseaux de surveillance et de sécurité, et même, pour les réseaux mobiles. En outre, il existe des réseaux où les noeuds ne sont pas homogènes et ne sont pas uniformément déployés sur la zone de captage. Ce qu'on peut dire à partir de cette contrainte, largement adoptée dans les réseaux de capteurs, est que le calcul de l'énergie consommée par chaque action qu'effectue un noeud capteur, n'est pas une tâche pratique. Nous pourrions alors penser à améliorer notre modèle de telle sorte qu'il soit plus adéquat pour de tels types de réseaux de capteurs.

En conclusion, du travail important reste à faire afin de fournir un cadre formel permettant une meilleure vérification des propriétés de la technologie réseaux de capteurs, notamment dans le cadre du routage économe en ressources énergétiques.

Bibliographie

- [1] Abdolreza Abhari and Ahmad Hosseingholizadeh. A neural network approach for wireless sensor network power management. *Article*, 30 August 2009.
- [2] A. Aissani and M. Ioualalen. Les symétries dans les réseaux de petri stochastiques (rdps). construction du graphe symbolique. *Revue française d'automatique, d'informatique et de recherche opérationnelle (RAIRO)*, 34(2) :237–249, 2000.
- [3] Jamal N. Al-karaki and Ahmed E. Kamal. Routing techniques in wireless sensor networks : A survey. *IEEE Wireless Communications*, 11 :6–28, 2004.
- [4] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126 :183–235, 1994.
- [5] Isaac Amundson and Xenofon D. Koutsoukos. A survey on localization for mobile wireless sensor networks. In *2nd International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT)*, September 2009.
- [6] Isabelle Augé-Blum and Thomas Watteyne. Proposition of a hard real-time mac protocol for wireless sensor networks. *Modeling, Analysis, and Simulation of Computer Systems, 13th IEEE International Symposium on*, 0 :533–532, 2005.
- [7] François Baccelli. Internet : modéliser le trafic pour mieux le gérer. *INRIA (Institut national de recherche en informatique et automatique) et École Normale Supérieure (Département d'informatique), Paris*, pages 75–79, December 2005.
- [8] J. C. M. Baeten. A brief history of process algebra. Technical report, Theor. Comput. Sci, 2004.
- [9] Pubali Banerjee. *Performance and security measure of clustering protocols for sensor networks*. PhD thesis, Iowa State University, 2007.
- [10] Antoine Belconde. *Modélisation de la détection de présence humaine*. PhD thesis, Institut PRISME, Université d'Orléans, 28 Septembre 2010.
- [11] Fatima Zohra Benhamida. Tolérance aux pannes dans les réseaux de capteurs sans fil. Magister thesis, ESI, Algérie, 2009.
- [12] Amel Berrachedi, Amina Diarbakirli, and Lyes Khelladi. Sécurisation d'un protocole de routage hiérarchique leach dans les réseaux de capteurs sans fil. Projet de fin d'étude, ESI, Algérie, June 2009.

-
- [13] Kamal Beydoun. *Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs*. PhD thesis, L'U.F.R des sciences et techniques de l'université de Franche-Comte, France, December 2009.
- [14] Chatschik Bisdikianb, David J. Thornleya, and Duncan F. Gilliesa. Using stochastic process algebra models to estimate the quality of information in military sensor networks. In *Proceedings Vol. 6965 : Modeling and Simulation for Military Operations III, Dawn A. Trevisani, Editors, 696507*, pages 1–10, April 2008.
- [15] Antonino Biundo, Antonio Puliafito, Luca Paladina, and Marco Scarpa. Artificial intelligence and synchronization in wireless sensor networks. *Journal of networks*, 4(6) :382–391, August 2009.
- [16] Patricia Bouyer, Serge Haddad, and Pierre-Alain Reynier. Timed Petri nets and timed automata : On the discriminating power of Zeno sequences. In Michele Buglesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06) — Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 420–431, Venice, Italy, July 2006. Springer.
- [17] Patricia Bouyer and François Laroussinie. Vérification par automates temporisés. In Nicolas Navet, editor, *Systèmes temps-réel 1 : techniques de description et de vérification*, pages 121–150. Hermès, June 2006.
- [18] Yacine Challal. Réseaux de capteurs sans fil, Novembre 2008.
- [19] Najla Chamseddine. *Analyse quantitative paramétrée d'automates temporisés probabilistes*. PhD thesis, Ecole Normale Supérieure ENS, France, September 2009.
- [20] Anantha Chandrakasan, Hari Balakrishnan, and Wendi Rabiner Heinzelman. Energy-efficient communication protocol for wireless microsensor networks. *Proceedings of the 33rd Hawaii International Conference on System Sciences*, 2 :1–10.
- [21] C.-F. Chiasserini and M. Garetto. Modeling the performance of wireless sensor networks. In *IEEE Infocom*, 2004.
- [22] Annie Choquet-Geniet. *Les réseaux de Petri - Un outil de modélisation : cours et exercices corrigés*. Dunod, Mars 2006.
- [23] Kourosch Dadashtabar, Mojtaba Hoseini, Neda Enami, and Reza Askari Moghadam. Neural network based energy efficiency in wireless sensor networks : a survey. *International Journal of Computer Science and Engineering Survey (IJCSES)*, 1(1) :39–55, August 2010.
- [24] Cristian Ene. *Un modèle formel pour les systèmes mobiles à diffusion*. PhD thesis, Université de la Méditerranée Aix-Marseille II, France, December 2001.

-
- [25] Emmanuel Fleury. *Les automates temporisés avec mises à jour*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, December 2002.
- [26] Dan Floroian, Dan Puiu, and Florin Moldoveanu. Petri nets and agents to supervisory control of complex environment. *Bulletin of the Transilvania University of Brasov*, 3(52) :267–276, 2010.
- [27] Jackson Francomme, Karen Godary, and Thierry Val. Validation formelle d’un mécanisme de synchronisation pour réseaux sans fil. In *CFIP’2009*, October 2009.
- [28] Antoine Gallais. *Ordonnancement d’activité dans les réseaux de capteurs : l’exemple de la couverture de surface*. PhD thesis, Université des Sciences et Technologies de Lille, France, June 2007.
- [29] Vashti Galpin. Modelling network performance with a spatial stochastic process algebra. In *Proceedings of the International Conference on Advanced Information Networking and Applications*, pages 41–49, June 2010.
- [30] Hubert Garavel. Défense et illustration des algèbres de processus. In *Ecole d’été Temps Réel*, September 2003.
- [31] Geneviève Gauthier. Les chaînes de markov : Modèles probabilistes et stochastiques de la gestion, January 2011. Service de l’enseignement des méthodes quantitatives de gestion, HEC Montréal.
- [32] Apolinar González, Alfons Crespo, Diego Martínez, Francisco Blanes, José Simo, and Raúl Aquino. Formal specification and design techniques for wireless sensor and actuator networks. 11(1) :1059–1077, January 2011.
- [33] Charles M. Grinstead and Laurie J. Snell. *Chapter 11 : Markov chains*, pages 405–470. American Mathematical Society, July 1997.
- [34] Ashish Gupta. *Empirical Analysis of Wireless Sensor Networks*. PhD thesis, ‘Ecole Doctorale EDITE, Paris, September 2010.
- [35] Osman Guzide, Seungyun Guzide, and Seth Cook. Towards an optimal network topology in wireless sensor networks. *Seattle University Journal of Undergraduate Research (SUJUR)*, Vol.1 :55–60, 2009.
- [36] Elyes Ben Hamida. *Modélisation stochastique et simulation des réseaux sans fil multi-sauts*. PhD thesis, INSA Lyon, France, September 2009.
- [37] Abdelhakim Hamzi. Plateforme basée agents pour l’aide à la conception et la simulation des réseaux de capteurs sans fil. Magister thesis, INI, Algérie, 2007.
- [38] Daojing He, Hejiao Huang, Lin Cui, and Maode Ma. Design and verification of enhanced secure localization scheme in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(7) :1050–1058, July 2009.

-
- [39] Hao JIANG, Xiao hui HU, and Xue na ZENG. Formal modeling and analysis approach of wireless sensor network. *Computer applications*, 30(7) :1722–1724, July 2010.
- [40] Rahim KACIMI. *Techniques de conservation d'énergie pour les réseaux de capteurs sans fil*. PhD thesis, Institut National Polytechnique de Toulouse, France, September 2009.
- [41] Krasimira Kapitanova and Sang H. Son. Event detection in wireless sensor networks. Paper, University of Virginia, USA, 2010.
- [42] Fadila Khadar and David Simplot-Ryl. Modélisation de protocoles mac pour réseaux de capteurs à l'aide de chaînes de markov. In *Colloque Francophone sur l'Ingénierie des Protocoles CFIP'08*, March 2008.
- [43] Lyes Khelladi. Les réseaux de capteurs : état de l'art. Magister thesis, USTHB-CERIST, Algérie, 2004.
- [44] Jong-Moon Kim. Applying dynamic markov random fields for sensor data analysis. *MIT Undergraduate Research Journal (MURJ)*, 13 :30–36, Spring 2006.
- [45] Jean-Claude Konig, Mohamed Benmohamed, and Samia Chelloug. Prédiction de la mobilité pour un routage efficace en energie dans les réseaux de capteurs mobiles. In *JDIR'09 : 10èmes Journées Doctorales en Informatique et Réseaux*, pages 139–143, 2009.
- [46] Helia Kousha, Ladan Darougaran, and Saeid Bahanfar. Neural networks for error detection and data aggregation in wireless sensor networks. *International Journal of Computer Science Issues (IJCSI)*, 8(3) :287–293, September 2011.
- [47] Andrea Kulakov and Danco Davcev. Tracking of unusual events in wireless sensor networks based on artificial neural-network algorithms. *Information Technology : Coding and Computing (ITCC)*, 2 :534–539, 2005.
- [48] Deepa Kundur and William Luh. Distributed privacy for visual sensor networks via markov shares. In *Proc. 2nd IEEE Workshop on Dependability and Security in Sensor Networks and Systems*, pages 23–34, April 2006.
- [49] H.L Leung, K.L. Man, T. Krilavièius, and Th. Vallee. Tepawsn : A formal analysis tool for wireless sensor networks. *International Journal of Research and Reviews in Computer Science (IJRRCS)*, 1(1) :24–26, August 2010.
- [50] Brian N. Levine, Jim Kurose, and Jim Partan. A survey of practical issues in underwater networks. In *Proc. ACM WUWNet*, pages 17–24, 2006.
- [51] Juzheng Li, Lily Changand, Sol M. Shatz, and Xudong He. Applying a nested petri net modeling paradigm to coordination of sensor networks with mobile agents.

-
- International Workshop on Petri Nets and Distributed Systems (PNDS 2008)*, Xi'an, China, June 2008.
- [52] Nan Liu and Sennur Ulukus. Optimal distortion-power tradeoffs in sensor networks : Gauss-markov random processes. *Journal : Computing Research Repository - CORR*, (cs.IT/0604040) :6, April 2006.
- [53] Yu Liu, Yuanli Cai, and Zhenhua Yu. Design of an energy-efficient distributed multi-level clustering algorithm for wireless sensor networks. *Wireless Communications, Networking and Mobile Computing, 2008. (WiCOM '08). 4th International Conference on*, pages 1–4, November 2008.
- [54] Ahmed Louazani, Bouabdellah Kechar, Larbi Sekhri, and Mohamed-Faycal Khelfi. Energy efficient cross-layer mac protocol for wireless sensor networks. In *International Workshop on Verification and Evaluation of Computer and Communication Systems : VECoS 2008*, pages 1–11, July 2008.
- [55] Yan Luo and Jeffrey J.P. Tsai. A graphical simulation system for modeling and analysis of sensor networks. *Multimedia, International Symposium on*, 0 :474–482, December 2005.
- [56] Angelika Mader, Ansgar Fehnker, and Lodewijk van Hoesel. Modelling and verification of the lmac protocol for wireless sensor networks. *Lecture Notes in Computer Science (LNCS)*, 4591/2007 :253–272, 2007.
- [57] Taous Madi. Utilisation des modèles formels dans les systèmes mobiles. Magister thesis, USTHB, Algérie, 2010.
- [58] Imad Mahgoub and Jamil Ibriq. Cluster-based routing in wireless sensor networks : Issues and challenges. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems' 2004 (SPECTS' 04)*, pages 759–769, 2004.
- [59] Arati Manjeshwar and Dharma P. Agrawal. Teen : A routing protocol for enhanced efficiency in wireless sensor networks. In *Parallel and Distributed Processing Symposium., Proceedings 15th International*, pages 2009–2015, Center for Distributed and Mobile Computing, ECECS Department, University of Cincinnati, 2001.
- [60] Catello Di Martino. *Resiliency assessment of wireless sensor networks : a holistic approach*. PhD thesis, 'Federico II' University of Naples, Italy, December 2009.
- [61] Allan I. McInnes. Model-checking the flooding time synchronization protocol. *IEEE International Conference on Control and Automation, 2009. ICCA 2009*, pages 422–429, December 2009.

- [62] Houda Bel Mokadem. *Vérification des propriétés temporelles des automates programmables industriels*. PhD thesis, École Normale Supérieure De Cachan, September 2006.
- [63] Amy L. Murphy, Gian P. Picco, Luca Mottola, and Paolo Costa. Teenylime : Transiently shared tuple space middleware for wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Middleware for Sensor Networks (MIDSENS)*, November 2006.
- [64] Sebastian Nanz. *Specification and Security Analysis of Mobile Ad-Hoc Networks*. PhD thesis, Imperial College of Science Technology and Medicine, University of London, May 2006.
- [65] Frank Oldewurtel and Petri Mähönen. Neural wireless sensor networks. *International conference on Systems and Networks Communications 2006 (ICSNC'06)*, 2 :1–28, October 2006.
- [66] Beatriz Ontiveros, Hugo Rodríguez, Ismael Soto, Rolando Carrasco, and Rubén Carvajal. Using petri net for modeling and analysis of a encryption scheme for wireless sensor networks. In Pawel Pawlewski, editor, *Petri Nets : Application*, pages 307–320. In-Tech, April 2011.
- [67] Dalimir Orfanus, Flávio Rech Wagner, and Tales Heimfarth. Process algebra to model self-organizing behavior in wireless sensor networks. In *Proceedings of the International Conference on Ultra Modern Telecommunications (ICUMT)*, pages 1–6. IEEE, October 2009.
- [68] Dalimir Orfanus, Flávio Rech Wagner, and Peter Janacik. Locality-aware extension of π -calculus to model self-organizing behavior in massively distributed embedded systems. *Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2010 13th IEEE International Symposium on*, pages 29–38, June 2010.
- [69] Abdelraouf Ouadjaout. La fiabilité de dissémination dans les réseaux de capteurs sans fil. Magister thesis, USTHB, Algérie, 2008.
- [70] Claire Pagetti. Introduction aux automates temporels. [www.onera.fr/staff/claire-pagetti/docs/cours\(1\).pdf](http://www.onera.fr/staff/claire-pagetti/docs/cours(1).pdf), IN2-math, 2011.
- [71] Lynne E. Parker and YuanYuan Li. Intruder detection using a wireless sensor network with an intelligent mobile robot response. *Southeastcon, 2008. IEEE*, pages 37–42, April 2008.
- [72] Rajendra M. Patrikar and Sudhir G. Akojwar. Improving life time of wireless sensor networks using neural network based classification techniques with cooperative routing. *International journal of communications*, 2(1) :75–86, November 2008.

- [73] Rajendra M. Patrikar and Sudhir G. Akojwar. Neural network based classification techniques for wireless sensor network with cooperative routing. *12th WSEAS International Conference on COMMUNICATIONS*, 7(8) :433–438, July 2008.
- [74] Ai PENG, Cheng WANG, Lan HUANG, and Zhong yi WANG. Model and analysis of leach protocol based on petri net. *Computer applications*, 29(24) :1059–1067, May 2009.
- [75] James L. Peterson. Petri nets. *ACM Computing Surveys*, 9(3) :223–252, 1977.
- [76] Walter H. Pitts and Warren S. McCulloch. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5 :115–133, 1943.
- [77] G. V. Pless, L. Reznik, and T. A. Karim. Signal change detection in sensor networks with artificial neural network structure. In *The IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety (CIHSPS)*, pages 44–51, March 2005.
- [78] Cauligi S. Raghavendra and Stephanie Lindsey. Pegasus : Power-efficient gathering in sensor information systems. *IEEE In Aerospace Conference Proceedings*, 3 :1125–1130, 2002.
- [79] T. Razafindralambo and F. Valois. Performance evaluation of backoff algorithms in 802.11 ad-hoc networks. In *ACM PEWASUN*, pages 82–89. ACM Press, 2006.
- [80] Isabelle RIVALS. Les réseaux de neurones formels pour le pilotage de robots mobiles. *FLUX, revue de l'Association amicale Les Ingénieurs*, (178), Semtember-October 1996.
- [81] Matej Rohlik and Tomas Vanek. Model of DoS resistant broadcast authentication protocol in colored petri net environment. In *17th International Workshop on Systems, Signals and Image Processing : IWSSIP 2010*, pages 264–267, 2010.
- [82] Ludovic SAMPER. *Modélisations et Analyses de Réseaux de Capteurs*. PhD thesis, laboratoire VERIMAG, l'école doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique, France, April 2008.
- [83] G. Scorletti and G. Binet. Réseaux de petri. Cours EL401T2 Master, 105 pages, Université de Caen/Basse-Normandie, France, 2006.
- [84] Qian-Ming Shao, Qi-Wei Ge, Qian Zhu, and Song Chen. Modeling and performance analysis of wireless sensor network systems using petri nets. In *The 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC'2008)*, pages 1689–1692, 2008.
- [85] Ali Shareef and Yifeng Zhu. Energy modeling of processors in wireless sensor networks based on petri nets. In *ICPPW'08 : Proceedings of the 2008 International*

- Conference on Parallel Processing - Workshops, (Portland, Oregon, USA)*, pages 129–134, 2008.
- [86] Ali Shareef and Yifeng Zhu. Energy modeling of wireless sensor nodes based on petri nets. In *ICPP : Proceedings of the International Conference on Parallel Processing, USA*), 2010.
- [87] Anu Singh, Scott A. Smolka, and Xiaowan Huang. Using integer clocks to verify the timing-sync sensor network protocol. In *Proceedings of NFM 2010*, pages 77–86, April 2010.
- [88] Dezhen Song. Probabilistic modeling of leach protocol and computing sensor energy consumption rate in sensor networks. Technical report, CS Department, Texas A and M University, February 2005.
- [89] Christos Stergiou and Dimitrios Siganos. Neural networks. *Imperial College of London*, 4 :28, 1996.
- [90] Huei-Ru Tseng, Rong-Hong Jan, and Wu Yang. A robust user authentication scheme with self-certificates for wireless sensor networks. In *Security Comm. Networks*. Wiley InterScience, 2010.
- [91] Faculté d’électronique et d’informatique USTHB. Cours de modélisation et simulation. 2007.
- [92] Ganesh K. Venayagamoorthy and Raghavendra V. Kulkarni. Neural network based secure media access control protocol for wireless sensor networks. pages 1680–1687, June 2009.
- [93] Mehmet C. Vuran, Xiao Fu, Yuanli Cai, and Zhenhua Yu. A reliable energy-efficient multi-level routing algorithm for wireless sensor networks using fuzzy petri nets. *Sensors’2011*, 11(3) :3381–3400, March 2011.
- [94] Mehmet C. Vuran, Steve Goddard, and Yunbo Wang. Stochastic analysis of energy consumption in wireless sensor networks. In *7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON)*, pages 1–9, June 2010.
- [95] Thomas Watteyne. Proposition et validation formelle d’un protocole mac temps réel pour réseaux de capteurs linéaires sans fils. Master thesis, INSA et CITI, France, 2005.
- [96] Kamin Whitehouse, Krasimira Kapitanova, Jingyuan Li, John A. Stankovic, Sang H. Son, and Yafeng Wu. Run time assurance of application-level requirements in wireless sensor networks. In *IPSN’10*, pages 197–208, April 2010.

-
- [97] Liang Xuedong, Simon Tschirner, and Wang Yi. Model-based validation of qos properties of biomedical sensor networks. In *Proceedings of the 8th ACM international conference on Embedded software EMSOFT '08*, pages 69–78, 2008.
- [98] Bashir Yahya, Jalel Ben-Othman, Lynda Mokdad, and Serigne Diagne. Performance evaluation of a medium access control protocol for wireless sensor networks using petri nets. In *HET-NETs'2010*, pages 335–354, 2010.
- [99] Bülent Yener, Fikret Sivrikaya, and Malik Magdon-Ismaïl. Joint problem of power optimal connectivity and coverage in wireless sensor networks. *Journal Wireless Networks*, 13(4) :537–550, August 2007.
- [100] Yaser YOUSEF. *Routage pour la Gestion de l'Energie dans les Réseaux de Capteurs Sans Fil*. PhD thesis, Université de Haute Alsace, France, May 2011.
- [101] Liyang Yu, Neng Wang, and Xiaoqiao Meng. Real-time forest fire detection with wireless sensor networks. *International Conference on Wireless Communications, Networking and Mobile Computing'2005*, pages 1214–1217, December 2005.
- [102] Li Yun, M. Daneshmand, You Xiaohu, Yu Nan, Zhang Weiyi, and Zhao Weiliang. Enhancing the performance of leach protocol in wireless sensor networks. *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 223 – 228, June 2011.
- [103] Sajeh Zaïri. Stratégies de configuration pour l'optimisation de la consommation énergétique dans les réseaux de capteurs. In *Journée thématique PHC/ResCom*, September 2010.
- [104] Wassim Znaidi. *Analyse par model-checking de la durée de vie d'un réseau de capteurs*. PhD thesis, Laboratoire Verimag, Université Joseph Fourier, France, 2007.
- [105] Belhassen Zouari, Eric Niel, and Sajeh Zaïri. Coloured petri net model for the formal validation of sensor networks. In *First International Conference on Sensor Networks and Applications : SNA'2009*, November 2009.