



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE



Laboratoire de
Recherche en
Informatique de
Tlemcen

UNIVERSITE ABOU-BEKR BELKAID - TLEMCEN

THÈSE

Présentée à :

FACULTÉ DES SCIENCES – DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :
DOCTORAT EN SCIENCES
Spécialité: Informatique

Par :

BENGHENI Abdelmalek

Sur le thème

Utilisation de la Récolte Énergétique pour l'Amélioration de la Durée de Vie des Réseaux de Capteurs Sans Fil

Soutenue publiquement le **15/06/2019** à Tlemcen devant le jury composé de :

Pr. CHIKH Azeddine	Professeur	Université de Tlemcen	Président
Dr. BENMAMMAR Badr Eddine	MCA	Université de Tlemcen	Examineur
Pr. MEJDI Kaddour	Professeur	Université d'Oran1 (Ahmed Ben Bella)	Examineur
Dr. KHALFI Mohammed Fethi	MCA	Université De Sidi-Belabbes	Examineur
Pr. LAHFA Fedoua	Professeur	Université de Tlemcen	Directrice de thèse

*Laboratoire de Recherche en Informatique de Tlemcen (LRIT)
13000 Tlemcen - Algérie*

Remerciements

Cette thèse a pu être réalisée grâce à la bénédiction d'Allah le tout-puissant, vers lequel vont toutes les louanges et à l'aide de l'assistance de nombreuses personnes que je voudrais les remercier pour leur soutien.

Je tiens tout d'abord à remercier vivement Madame. **LAHFA Fedoua**, Professeur au département d'informatique de l'université Abou Bekr Belkaid de Tlemcen et directrice de laboratoire de recherche en informatique de Tlemcen (**LRIT**), pour la confiance et la liberté qu'elle m'a accordées en acceptant la direction scientifique de ce travail, auquel j'exprime ma profonde reconnaissance pour ces conseils et orientations, et de son efficacité certaine que je n'oublierai jamais.

J'aimerais ensuite exprimer toute ma reconnaissance et ma gratitude envers mon cher frère et collègue **Dr. BAMBRIK Ilyas**, pour son aide précieuse et continue, sa disponibilité et ses encouragements qui m'ont été très bénéfiques.

Un remerciement très chaleureux à Monsieur. **CHIKH Azeddine**, Professeur au département d'informatique de l'université Abou Bekr Belkaid de Tlemcen, qui nous fait l'honneur d'avoir d'accepter de présider le jury de cette thèse.

Je remercie également Messieurs **BENMAMMAR Badr Eddine.**, **MEJDI Kaddour.**, et **KHALFI Mohammed Fethi**. Pour le grand plaisir et honneur qu'ils me font en acceptant de consacrer leur temps pour examiner ce travail. Qu'ils trouvent ici l'expression de ma profonde reconnaissance.

Mes remerciements les plus sincères vont aux membres du laboratoire de traitement et transport de l'information (**L2TI**) de l'université paris 13 pour leur accueil chaleureux, leur esprit du groupe durant mes stages et en particulier à Monsieur **ACHIR Nadjib.**, pour ses remarques, ses conseils et ses orientations, qui m'ont beaucoup aidé.

Enfin, je tiens à remercier mes parents pour leurs soutiens qui m'ont été bien nécessaires durant mon travail de thèse et toutes les personnes qui ont contribué à l'aboutissement de ce travail.

*À mes chers parents Yamina et Mohamed,
Pour leur éducation, leurs patiences et leur soutien indéfectible*

*À ma chère épouse,
Pour sa patience et ses sacrifices*

À mes chères sœurs,

À mes chers frères,

À toute ma famille et ma belle-famille,

*À tous mes collègues du département Informatique,
De l'université d'IBN-KHALDOUN-TIARET*

*À tous mes collègues et mes enfants de l'association Kafil EL Yatim,
EL Wilaeya-Tiaret, ALgérie*

À tous ceux que j'aime,

Je dédie ce travail.

Abdelmalek. B

La récupération d'énergie est le processus qui consiste à obtenir de l'énergie à partir de l'environnement externe (Comme: la lumière solaire et artificielle, les mouvements vibratoires, la chaleur, le rayonnement électromagnétique, etc.) et de transformer cette énergie en une énergie électrique, directement utilisable ou emmagasinée afin de servir ultérieurement au fonctionnement d'appareils de petite taille. L'énergie récupérée est généralement très faible (de l'ordre du mJ) par rapport aux systèmes de récupération d'énergie à grande échelle tels que les fermes solaires ou encore les fermes d'éoliennes.

Cette récolte de l'énergie ambiante peut être utilisée pour alimenter de petits capteurs autonomes sans fil qui ont de petites batteries et qui sont déployés dans des endroits éloignés ou hostiles pour la détection d'événements. De ce fait, l'intégration des systèmes de la récolte énergétique est une solution prometteuse pour prolonger la durée de vie des capteurs et améliorer leurs performances.

Dans cette thèse, nous nous intéressons à cette problématique. L'utilisation de la technique de récupération énergétique pour les réseaux de capteurs sans fil. Plus précisément, l'objectif est d'adapter le fonctionnement du réseau aux cycles de recharge décharge des capteurs. Il s'agit d'une façon générale, de basculer entre deux états (veille et réveil) en se basant sur la technique duty-cycle (i.e., un capteur avec un niveau d'énergie inférieur sélectionne une période de veille plus longue pour collecter autant d'énergie que possible avant de passer à l'état de déchargement (réveil) où l'énergie déposée est consommée). En effet, puisque le réveil d'un capteur ne peut pas être estimé avec précision, du fait que le taux exact d'énergie récolté par le capteur fluctue dans le temps, il est très difficile de s'assurer de l'acheminement des paquets. De plus, l'incertitude sur le temps qu'il faut à un nœud afin de récolter assez d'énergie avant qu'il puisse fonctionner à nouveau fait en sorte que les solutions de planification de veille réveil d'un capteur proposées dans la littérature sont inefficaces et donc inutilisables.

Par l'introduction d'une politique de seuil d'énergie et l'utilisation de l'énergie restante au niveau de la batterie d'un capteur, les périodes de réveil et de veille sont réglementées afin de diminuer le duty-cycle de chaque capteur dans le réseau tout en assurant un équilibre entre la consommation d'énergie et la capacité de récupération d'énergie. La politique de seuil d'énergie proposée comprend deux phases (premièrement, la phase de commutation entre trois états possibles pour chaque capteur pour réguler sa période de réveil et d'assurer un équilibre de charge dans le réseau, la seconde est le calcul de la durée de la période de veille afin d'optimiser le duty-cycle et le rendre dynamique).

Les principaux résultats obtenus à l'aide de l'intégration de cette politique dans les différents types de communications utilisées entre les capteurs dans ce type de réseau, sont la possibilité de minimiser le nombre de collisions, le nombre maximal de retransmissions et la contention sur le canal dans le réseau. De ce fait, l'utilisation de cette politique nous a permis de prolonger la durée de vie du réseau et d'améliorer ses performances.

En conséquence, l'augmentation de la durée de vie d'un réseau de capteurs sans fil est un travail compliqué, lourd, et qui prend beaucoup de temps. Bien que, expérimentalement, l'amélioration de la gestion de l'énergie dans les réseaux de capteurs sans fil à récupération d'énergie est nettement bénéfique pour surmonter certaines lacunes de plusieurs travaux existants et de rendre cette gestion plus fondamentale et plus pratique.

Mots clés: Réseau de capteurs sans fil; La récupération d'énergie; duty-cycle; La durée de vie du réseau de capteurs sans fil.

The energy harvesting is the process of obtaining energy from the external environment (e.g., solar and artificial light, vibratory motion, heat, electromagnetic radiation, etc.) and transforming this energy into electrical energy, directly used or stored for use in the operation of small devices. The harvested energy is generally very low (in the mJ range) compared to large-scale energy harvesting systems such as solar farms or wind farms.

This harvest of ambient energy can be used to power the small wireless autonomous sensors that have small batteries and are deployed in remote or hostile locations for event detection. Therefore, the integration of energy harvesting systems is a promising solution to extend the sensor lifetime and improve its performances.

In this thesis, we are interested in this problematic. The use of the energy harvesting technique for wireless sensor networks. More specifically, the objective is to adapt the operation of the network to the recharging/discharging cycles of the sensors. It's about a general way of switching between two states (sleep and wake-up) based on the duty-cycle technique (i.e., a sensor with a lower energy level selects a longer sleep period to harvest as much energy as possible before going to the unloading state (wake-up) where the deposited energy is consumed). Indeed, since the wake-up of a sensor cannot be accurately estimated, the fact that the exact rate of harvested energy by the sensor fluctuates over time, it is very difficult to ensure the routing of the packets. In addition, the uncertainty over the time it takes for a node to harvest enough energy before it can work again means that a sensor's sleep/wake-up planning solutions are proposed in the literature are unusable.

By introducing an energy threshold policy and using the remaining energy in the battery of a sensor, the wake-up and sleep periods are regulated to decrease the duty cycle of each sensor in the network while ensuring a balance between energy consumption and the energy harvesting capacity. The proposed energy threshold policy comprises two phases (firstly, the switching phase between three possible states for each sensor to regulate its wake-up period and ensure load balance in the network and the second, is the calculation of the duration of the sleep period to optimize the duty cycle and make it dynamic).

The main results obtained through the integration of this policy in the different types of communications used between the sensors in this type of network, are the possibility of minimizing the number of collisions, the maximum number of retransmissions and the contention on the channel in the network. As a result, the use of this policy has allowed us to extend the life of the network and improve its performance.

Consequently, increasing the lifetime of a wireless sensor network is a complicated heavy work, and that takes a lot of time. Although the experience obtained by proposing the implementing the improvement of energy management in energy harvesting wireless sensor networks is clearly beneficial in overcoming some of the shortcomings of several existing works and making this management more fundamental and more practical.

Keywords: Wireless sensor network; Energy harvesting; duty-cycle; Wireless sensor network lifetime.

ملخص

إن حصاد الطاقة هي عملية يتم فيها جمع الطاقة من مصادر خارجية (البيئة الخارجية) (مثل: ضوء الشمس، الضوء الإصطناعي، الحركة الإهتزازية، الحرارة والإشعاع الكهرومغناطيسي، وغيرها...) وتحويل هذه الطاقة إلى طاقة كهربائية يمكن إستخدامها مباشرة أو تخزينها قصد صرفها في تشغيل أجهزة الصغيرة.

عموما ما تكون هذه الطاقة المحصودة منخفضة (في نطاق مج) مقارنة بأنظمة حصاد الطاقة المتواجدة على مجال شاسع مثل المزارع الشمسية أو مزارع الرياح. كما يمكن إستخدام هذا الحصاد من الطاقة المحيطة لتشغيل أجهزة مستشعرات لاسلكية صغيرة مستقلة ذات بطاريات صغيرة أين يتم نشرها في مواقع بعيدة أو معادية من أجل الكشف عن الأحداث. ونتيجة لذلك، يعد إدراج أنظمة حصاد الطاقة حلاً واعدًا لإطالة عمر المستشعر وتحسين الأداء.

في هذه الرسالة، نحن مهتمون بهذه الإشكالية. إستخدام تقنية حصاد الطاقة لشبكات الاستشعار اللاسلكية. وبشكل أكثر تحديداً، فإن الهدف هو كيفية تكييف تشغيل الشبكة مع دورات التفريغ والشحن الخاصة بالمستشعرات. وبشكل عام فهذا التكييف ماهو إلا طريقة الإنتقال بين حالتين (النوم والاستيقاظ) بناءً على تقنية دورة العمل (أي أن المستشعر ذو مستوى طاقة منخفض يختار فترة نوم أطول لتجميع أكبر قدر ممكن من الطاقة قبل الذهاب إلى حالة التفريغ (الاستيقاظ) حيث يتم استهلاك الطاقة المودعة). في الواقع، بما أن وقت إستيقاظ المستشعر لا يمكن تقديره بدقة، لأن معدل الطاقة الدقيق الذي يجمعه المستشعر يتغير بمرور الوقت، فمن الصعب للغاية ضمان توجيه الحزم. بالإضافة إلى ذلك، فإن عدم التيقن من الوقت الذي تستغرقه العقدة لحصاد ما يكفي من الطاقة قبل أن تتمكن من العمل مرة أخرى يعني أن حلول جدولة فترات النوم والاستيقاظ لجهاز استشعار المقترحة من قبل غير قابلة للإستخدام.

من خلال إدخال سياسة عتبة الطاقة وإستخدام الطاقة المتبقية في بطارية جهاز الاستشعار، يتم تنظيم فترات اليقظة والنوم لتقليل دورة التشغيل لكل جهاز استشعار في الشبكة مع ضمان التوازن بين استهلاك الطاقة وقدرة حصاد الطاقة.

ملخص

تستخدم سياسة الحد الأدنى المقترحة للطاقة مرحلتين (أولهما، مرحلة التبديل بين ثلاث حالات محتملة لكل جهاز استشعار لتنظيم فترة الاستيقاظ وضمان توازن الحمل في الشبكة والثانية، تتمثل في حساب مدة فترة النوم لتقليل من دورة العمل وجعلها ديناميكية).

النتائج الرئيسية المحصل عليها من خلال إدراج هذه السياسة في أنواع الاتصالات المختلفة المستعملة بين أجهزة الاستشعار في هذا النوع من الشبكات، هي القدرة على تقليل من: عدد حالات التصادم، الحد الأقصى لعدد عمليات إعادة الإرسال، والتنافس على القناة في الشبكة. ونتيجة لذلك، سمح لنا استخدام هذه السياسة بتمديد عمر الشبكة وتحسين أدائها.

وخلاصة لذلك ، فإن زيادة عمر شبكة الإستشعار اللاسلكية هي مهمة معقدة وتستهلك الكثير من الوقت. وعلى الرغم من أن الخبرة المكتسبة في تطبيق تحسين إدارة الطاقة في شبكات أجهزة الاستشعار اللاسلكية لحصاد الطاقة مفيدة بشكل واضح في التغلب على بعض أوجه القصور في العديد من الأعمال القائمة وجعل هذه الإدارة أساسية وأكثر عملية.

الكلمات المفتاحية: شبكة الإستشعار اللاسلكية؛ حصاد الطاقة؛ دورة العمل؛ عمر شبكة الإستشعار اللاسلكية.

Liste des Publications Personnelles

Articles publiés dans des revues scientifiques:

Abdelmalek Bengheni, Fedoua Didi, Ilyas Bambrik: *EEH-EHWSN: Enhanced Energy Management Scheme in Energy Harvesting Wireless Sensor Networks*. *Wireless Networks (The Journal of Mobile Communication, Computation and Information Springer, ISSN 1022-0038(Print) 1572-8196 (Online))* 09/03/2018; DOI: 10.1007/s11276-018-1701-8. IF: 1.981 (Indexed Thomson Reuters, Scopus...)

Abdelmalek Bengheni, Fedoua Didi: *Evaluating the Performance of Asynchronous MAC Protocols for Wireless Sensor Networks*. *International Journal of Wireless and Mobile Computing*, Inderscience, ISSN 1741-1084 (Print) 1741-1092 (Online), (2019). (Indexed Scopus, Compendex, DBLP...)

Articles de conférences:

Abdelmalek Bengheni, Fedoua Didi, Ilyas Bambrik: *Energy-harvested management mechanism for wireless sensor networks*. the 5th International Conference on Electrical Engineering - Boumerdes (ICEE-B), Algeria, October 29th - 31th, 2017 (Indexed IEEE Xplore)

Abdelmalek Bengheni, Fedoua Didi, Ilyas Bambrik: *Energy-saving comparison of asynchronous MAC protocols for wireless sensor networks*. the International Conference on Mathematics and Information Technology (ICMIT), Adrar, Algeria, December 04th - 05th, 2017 (Indexed IEEE Xplore)

Abdelmalek Bengheni, Fedoua Didi, Ilyas Bambrik, Adda Boualem: *A Preamble Sampling Scheme based MAC Protocol for Energy Harvesting Wireless Sensor Networks*. the 3rd Conference on Computing Systems and Applications (CSA), Ecole Militaire Polytechnique, Algiers, Algeria, April 24th - 25th, 2018 (Indexed ACM, DBLP...)

Articles de Workshops:

Abdelmalek Bengheni, Fedoua Didi: *Improvement of the Lifetime of Wireless Sensor Network: state of the Art*. The second International Workshop on Mathematics and Computer Science IWMCS' 2014, Tiaret-Algeria, December 1- 2, 2014

Table des matières

Introduction générale.....	01
1 Contexte et motivations.....	01
2 Objectifs de la thèse.....	02
3 Contributions.....	03
4 Organisation de la thèse.....	05
Partie I : État de l'art et contexte	
1 Les Réseaux de Capteurs Sans Fil	
1.1 Introduction.....	07
1.2 Les réseaux <i>ad hoc</i>	07
1.3 Les réseaux de capteurs sans fil.....	08
1.3.1 Architecture d'un nœud capteur.....	08
1.3.2 Architecture d'un <i>RCSF</i>	11
1.3.3 Les types de base d'un <i>RCSF</i>	13
1.3.4 Les besoins des applications de <i>RCSF</i>	14
1.3.4.1 L'efficacité énergétique.....	14
1.3.4.2 Le passage à l'échelle (La scalabilité).....	14
1.3.4.3 La bande passante limitée.....	14
1.3.4.4 La latence.....	14
1.3.4.5 Le débit.....	15
1.3.4.6 L'équité dans le partage du canal.....	15
1.3.4.7 Tolérance aux pannes.....	15
1.3.4.8 Auto-configuration.....	16
1.3.4.9 Taille et coût réduit.....	16
1.3.5 Domaines d'applications des <i>RCSF</i>	16
1.3.5.1 Les applications environnementales.....	17
1.3.5.2 Les applications militaires.....	18
1.3.5.3 Les applications médicales.....	18
1.3.5.4 Les applications liées à la sécurité.....	19
1.3.5.5 Les applications industrielles.....	19
1.3.6 La pile protocolaire dans les <i>RCSF</i>	19
1.3.6.1 La couche physique.....	20
1.3.6.2 La couche liaison de données.....	20
1.3.6.3 La couche réseau.....	21

1.3.6.4	La couche transport.....	21
1.3.6.5	La couche application.....	21
1.3.6.6	Plan de gestion de l'énergie.....	22
1.3.6.7	Plan de gestion de mobilité.....	22
1.3.6.8	Plan de gestion des tâches.....	22
1.3.7	La consommation d'énergie dans les <i>RCSF</i>	22
1.3.7.1	La consommation d'énergie d'un nœud capteur.....	23
1.3.7.2	Les sources des pertes d'énergie.....	24
1.3.8	La couverture dans un <i>RCSF</i>	25
1.3.9	La connectivité dans un <i>RCSF</i>	26
1.3.10	La durée de vie d'un <i>RCSF</i>	26
1.4	Conclusion.....	27
2	Les protocoles <i>MAC</i> et la conservation d'énergie dans les <i>RCSF</i>	
2.1	Introduction.....	28
2.2	La sous couche <i>MAC</i>	28
2.3	La technique de cycle d'activité (<i>Duty-cycle</i>).....	29
2.3.1	Les protocoles <i>Sleep/Wakeup</i> (<i>Veille/Réveil</i>).....	29
2.3.2	La classification des protocoles <i>MAC</i> de faible <i>Duty-cycle</i>	30
2.3.2.1	Les protocoles <i>MAC</i> basés sur la contention.....	31
2.3.2.2	Les protocoles <i>MAC</i> basés sur la planification.....	38
2.3.2.3	Les protocoles <i>MAC</i> basés sur l'interrogation du canal.....	40
2.3.2.4	Les protocoles <i>MAC</i> hybrides.....	43
2.3.3	La comparaison de différents protocoles <i>MAC</i> pour <i>RCSF</i>	46
2.4	D'autres techniques de conservation d'énergie.....	47
2.4.1	Les techniques dirigées par les données.....	47
2.4.1.1	L'acquisition des données.....	47
2.4.1.2	Traitement et transmission des données.....	48
2.4.2	Les techniques basées sur la mobilité.....	49
2.5	Conclusion.....	50
3	Les protocoles <i>MAC</i> pour les <i>RCSF</i> à récupération d'énergie	
3.1	Introduction.....	51
3.2	Définition de la récupération énergétique.....	51
3.3	Les nœuds de capteur à récupération d'énergie.....	51
3.3.1	Architecture du système de récupération d'énergie.....	52
3.3.1.1	Architecture <i>Récupérer-dépenser</i>	52
3.3.1.2	Architecture <i>Récupérer-recharger-dépenser</i>	52
3.4	Les sources d'énergie pour la récupération énergétique.....	53

3.5	Les techniques de la récupération énergétique.....	53
3.5.1	Récupération de l'énergie mécanique.....	53
3.5.2	Récupération de l'énergie photovoltaïque.....	54
3.5.3	Récupération de l'énergie thermique.....	54
3.5.4	Récupération de l'énergie biochimique.....	55
3.5.5	Récupération de l'énergie acoustique.....	55
3.5.6	Récupération de l'énergie des signaux radio fréquence (<i>RF</i>).....	55
3.5.7	Récupération de l'énergie à partir du corps humain.....	55
3.6	Quelques applications des nœuds de capteur à récupération d'énergie.....	56
3.6.1	Suivi des animaux.....	56
3.6.1.1	<i>ZebraNet</i>	56
3.6.1.2	<i>TurtleNet</i>	56
3.6.2	La surveillance de la qualité de l'air.....	57
3.6.3	La surveillance de la qualité de l'eau.....	57
3.6.4	La surveillance de la santé.....	58
3.6.5	La surveillance des catastrophes.....	58
3.7	Les motivations pour les protocoles <i>MAC</i> spéciaux pour les <i>RCSF-RE</i>	59
3.7.1	Le principe de conception d'un <i>RCSF-RE</i>	59
3.7.2	L'adaptabilité de la technique duty-cycling.....	59
3.7.3	Les capacités de la récolte.....	60
3.7.4	Fonctionnement neutre en énergie.....	60
3.7.5	Les caractéristiques de l'énergie.....	62
3.7.6	Les profils de charge variables.....	62
3.8	Les protocoles <i>MAC</i> pour les <i>RCSF</i> à récupération d'énergie.....	62
3.8.1	Les protocoles <i>MAC</i> d'interrogation probabiliste pour les <i>RCSF-RE</i>	63
3.8.2	Les protocoles <i>MAC</i> d'interrogation non probabiliste pour les <i>RCSF-RE</i>	65
3.9	Analyse théorique de différents schémas <i>MAC</i> pour <i>RCSF-RE</i>	73
3.10	Conclusion.....	76

Partie II : Contributions

4	Comparaison de la conservation d'énergie des protocoles <i>MAC</i> asynchrones pour les <i>RCSF</i>	
4.1	Introduction.....	77
4.2	La simulation.....	77
4.3	Les modèles de la consommation d'énergie.....	78
4.3.1	Le modèle de la consommation d'énergie du protocole <i>BMAC</i>	78
4.3.2	Le modèle de la consommation d'énergie du protocole <i>XMAC</i>	80
4.3.3	Le modèle de la consommation d'énergie du protocole <i>RIMAC</i>	80

4.4	Les outils de simulation et l'implémentation détaillée.....	81
4.4.1	La machine à états finis du nœud émetteur et récepteur avec <i>XMAC</i>	82
4.4.2	La machine à états finis du nœud émetteur et récepteur avec <i>RIMAC</i>	83
4.5	Les paramètres de simulation.....	85
4.6	Résultats et Discussion.....	86
4.7	Conclusion.....	89
5	<i>EEM-EHWSN</i> : Un schéma de gestion de l'énergie amélioré dans les <i>RCSF-RE</i>	
5.1	Introduction.....	90
5.2	Architecture d'un nœud de capteur utilisée.....	90
5.3	Schéma de communication fondamental utilisé dans <i>EEM-EHWSN</i>	91
5.4	Le modèle de la consommation d'énergie et de la latence.....	93
5.4.1	Le modèle de la consommation d'énergie.....	94
5.4.2	Le modèle de la latence.....	95
5.5	La politique de seuil d'énergie proposée dans <i>EEM-EHWSN</i>	95
5.5.1	Mécanisme de gestion de l'énergie récupérée pour les <i>RCSF</i>	95
5.5.1.1	Formulation du problème d'optimisation.....	95
5.5.1.2	Les fonctions de base du <i>EH2M</i>	96
5.5.2	Aperçu de la politique de seuil d'énergie.....	97
5.5.2.1	L'étape de commutation.....	98
5.5.2.2	L'étape de calcul de la durée de la période de veille.....	98
5.5.3	Le mécanisme proposé pour le calcul de la période de veille.....	99
5.5.3.1	L'utilisation de la technique de commutation proposée en état de réveil.....	99
5.5.3.2	La procédure proposée pour calculer la durée de la période de veille.....	99
5.5.3.3	Les fonctions de base du <i>EEM-EHWNS</i>	100
5.6	La simulation.....	103
5.6.1	Les outils de simulation et les détails de l'implémentation.....	105
5.6.1.1	La machine à états finis du nœud émetteur et récepteur avec <i>EEM-EHWNS</i>	106
5.6.2	Les paramètres de simulation.....	108
5.7	Évaluation des performances.....	110
5.8	Conclusion.....	114
6	<i>PS-EHWSN</i> : Un protocole <i>MAC</i> basé sur un schéma d'échantillonnage de préambule pour les <i>RCSF-RE</i>	
6.1	Introduction.....	116
6.2	Architecture d'un nœud de capteur utilisée.....	116
6.3	Schéma de communication fondamental utilisé dans <i>PS-EHWSN</i>	117

6.4	Le modèle de la consommation d'énergie et de la latence.....	118
6.4.1	Le modèle de consommation d'énergie.....	119
6.4.2	Le modèle de la latence.....	119
6.5	Les fonctions de base du <i>PS-EHWSN</i>	120
6.6	La simulation.....	122
6.6.1	La machine à état finis du nœud émetteur et récepteur avec <i>PS-EHWNS</i>	122
6.6.2	Les paramètres de simulation utilisés.....	124
6.7	Évaluation des performances.....	125
6.8	Conclusion.....	129
 Conclusion générale et perspectives.....		 130
1	Conclusion générale.....	130
2	Les perspectives.....	132
 Annexe A.....		 134
Bibliographie		138

Liste des figures

1.1	Architecture d'un nœud capteur.....	08
1.2	Exemples sur les modèles de capteurs sans fil.	10
1.3	Architecture d'un <i>RCSF</i>	12
1.4	Architecture plate d'un <i>RCSF</i>	12
1.5	Architecture hiérarchique à base de clusters d'un <i>RCSF</i>	12
1.6	Quelques nœuds capteurs (mobiles et stationnaires).....	13
1.7	Quelques exemples des domaines d'applications des <i>RCSF</i>	16
1.8	Quelques images des domaines d'applications des <i>RCSF</i>	17
1.9	La pile protocolaire d'un <i>RCSF</i>	20
1.10	Consommation de l'énergie électrique par un nœud capteur.....	23
1.11	Les différents rayons (de couverture et de communication) d'un capteur.....	25
2.1	La structure de la couche liaison de données.....	29
2.2	La classification des protocoles <i>MAC</i> selon le type d'accès au canal.....	31
2.3	Les différents seuils utilisés par le mécanisme <i>CCA</i>	32
2.4	Illustration du problème de la station cachée/exposée.....	33
2.5	Les deux périodes « d'activation et de veille » périodique du <i>S-MAC</i>	35
2.6	La formation des clusters virtuels avec <i>S-MAC</i>	35
2.7	Les deux périodes « d'activation et de veille » périodique du <i>S-MAC</i> et <i>T-MAC</i>	37
2.8	Arbre de conversion convergée avec correspondance, emplacements <i>DMAC</i> décalés.....	37
2.9	Les deux périodes « d'accès aléatoires et d'accès planifiés » dans <i>TRAMA</i>	39
2.10	La transmission des données avec le protocole <i>BMAC</i>	41
2.11	La transmission des données avec le protocole <i>XMAC</i>	41
2.12	La transmission des données avec le protocole <i>RIMAC</i>	42
2.13	Structure de slot avec préférence intégrée pour les propriétaires de slot.....	43
2.14	La transmission des données avec le protocole <i>SCP-MAC</i>	44
2.15	L'administration du goulot d'étranglement dans <i>Funneling-MAC</i>	45
2.16	Une trame <i>CSMA/CA</i> est réservée entre deux planifications de trame <i>TDMA</i> dans <i>Funneling-MAC</i>	45
3.1	Les deux Architectures d'un système de récupération d'énergie.....	52
3.2	Quelques d'exemples de techniques de récupération d'énergie.....	54
3.3	Exemples de tendances de consommation d'énergie.....	61

3.4	Exemple de processus d' <i>EH-MAC</i>	64
3.5	Exemple de processus de <i>MTPP</i> avec un <i>RCSF</i> à 3 niveaux.....	65
3.6	Illustration du fonctionnement du protocole <i>OD-MAC</i>	66
3.7	Illustration du fonctionnement du protocole <i>ERI-MAC</i>	67
3.8	Exemple de processus de transmission de données prioritaire dans <i>QAEE-MAC</i>	67
3.9	Le fonctionnement de la station de base pendant un temps de trame.....	68
3.10	Un exemple du protocole <i>EL-MAC</i>	69
3.11	Illustration du fonctionnement du protocole <i>LEB-MAC</i>	71
3.12	Graphe de décision du <i>DSR</i> et <i>DSP</i>	72
4.1	(a) un réseau de <i>XMAC</i> , (b) un réseau de <i>RIMAC</i>	82
4.2	Machine à états finis de <i>HostXMAC</i>	83
4.3	Machine à états finis de <i>HostRIMAC</i>	84
4.4	Énergie consommée moyenne pour les trois protocoles <i>MAC</i> asynchrones avec un Traffic Param = 0.5s.....	87
4.5	Énergie consommée moyenne pour les trois protocoles <i>MAC</i> asynchrones avec un Traffic Param = 1.5s.....	87
4.6	Énergie consommée moyenne pour les trois protocoles <i>MAC</i> asynchrones avec un Traffic Param = 2.5s.....	87
4.7	Énergie consommée moyenne pour trois protocoles <i>MAC</i> asynchrones avec un nombre de paquets envoyés = 500 et un nombre de nœuds = 100.....	87
5.1	Les composants d'un nœud capteur alimenté par un récupérateur d'énergie à partir d'une source externe.....	91
5.2	Un émetteur <i>E</i> attendra une balise <i>B</i> du récepteur <i>R</i> avant d'envoyer ses données et le double rôle de la balise d'accusé de réception des données dans la communication.....	92
5.3	Dans <i>EEM-EHWSN</i> , l'émetteur <i>E</i> envoie initialement une balise <i>Beacon-on-Request</i> et envoie un paquet de données au nœud <i>R</i> après la réception de la balise d'accusé de réception.....	92
5.4	Fonctions de base du mécanisme <i>EH2M</i>	97
5.5	Diagramme de transition d'état pour les trois modes possibles du nœud de capteur dans <i>EEM-EHWSN</i>	98
5.6	Fonctions de base du système <i>EEM-EHWSN</i> où chaque nœud de capteur est équipé d'un système de récupération d'énergie.....	102
5.7	Les métriques de performance. (a) Latence moyenne, (b) Taux de livraison, (c) Le Débit.....	103
5.8	Le réseau du schéma <i>EEM-EHWSN</i>	106
5.9	Machine à états finis de <i>HostEEMEHWSN</i>	106

5.10	Le taux de livraison des paquets. (a) paramètre de trafic = 1.5 s, (b) paramètre de trafic = 2.0 s.....	111
5.11	La latence moyenne des paquets. (a) paramètre de trafic = 1.5 s, (b) paramètre de trafic = 2.0 s.....	112
5.12	Le débit. (a) paramètre de trafic = 1.5 s, (b) paramètre de trafic = 2.0 s.....	113
6.1	Illustration du fonctionnement de notre protocole <i>PS-EHWSN</i>	117
6.2	Fonctions de base du protocole <i>PS-EHWSN</i> où chaque nœud de capteur est équipé d'un système de récupération d'énergie.....	121
6.3	Le réseau du protocole <i>PS-EHWSN</i>	122
6.4	Machine à états finis de <i>HostPSEHWSN</i>	123
6.5	Le taux de livraison des paquets. (a) Nombre de paquets envoyés = 500, (b) Nombre de paquets envoyés = 1000.....	126
6.6	La latence moyenne des paquets. (a) Nombre de paquets envoyés = 500, (b) Nombre de paquets envoyés = 1000.....	127
6.7	Le débit. (a) Nombre de paquets envoyés = 500, (b) Nombre de paquets envoyés = 1000.....	128
A.1	Les différents modules d'un module système (simple et composé).....	134
A.2	La structure d'un nœud et la structure du module <i>NIC</i>	136

Liste des tableaux

1.1	Les caractéristiques de quelques modèles de capteurs sans fil.....	11
2.1	La comparaison des différents protocoles <i>MAC</i> pour les <i>RCSF</i>	46
3.1	La comparaison des différents schémas <i>MAC</i> pour les <i>RCSF-RE</i>	75
4.1	Les termes utilisés pour les modèles de réseau et d'énergie.....	79
4.2	Les paramètres de simulation.....	85
5.1	Les termes utilisés pour les modèles : réseau, l'énergie et la latence dans <i>EEM-EHWSN</i>	93
5.2	Les paramètres de simulation à différentes couches pour l'implémentation du <i>EEM-EHWSN</i>	109
5.3	Les paramètres de simulation de la puissance énergétique et de la récolte d'énergie pour l'implémentation du <i>EEM-EHWSN</i>	109
6.1	Les termes utilisés pour les modèles : réseau, l'énergie et la latence dans <i>PS-EHWSN</i>	118
6.2	Les paramètres de simulation à différentes couches pour l'implémentation du <i>PS-EHWSN</i>	124
6.3	Les paramètres de simulation de la puissance énergétique et de la récolte d'énergie pour l'implémentation du <i>PS-EHWSN</i>	124

Table des acronymes

ADC	A nalog to D igital C onverter
BAN	B ody A rea N etworks
BMAC	B erkely M AC
CCA	C lear C hannel A ssessment
CDMA	C ode D ivision M ultiple A ccess
ChSim	C hannel S imulator
CSMA	C arrier S ense M ultiple A ccess
CSMA/CA	C arrier S ense M ultiple A ccess with C ollision A voidance
DTN	D elay T olerant N etwork
DMAC	D ata gathering M AC
EEM-EHWSN	E nhanced E nergy M anagement scheme in E nergy H arvesting W SN
EH2M	E nergy H arvested M anagement M echanism
ENO	E nergy N eutral O peration
FDMA	F requency D ivision M ultiple A ccess
FLAMA	F low- A ware- M AC
GPS	G lobal P ositioning S ystem
IEEE	I nstitute of E lectrical and E lectronics E ngineers
INI	I Nitialization
LLC	L ogical L ink C ontrol
LPL	L ow P ower L istening
MAC	M edium A ccess C ontrol
MANET	M obile A d-hoc N ETworks
MiXiM	M i X ed S i M ulator;
MF	M obility F ramework
NED	N etwork D escription

NIC	N etwork I nterface C ard
OMNeT++	O bjective M odular N etwork T estbed in C++;
OSI	O pen S ystem I nterconnection
PHY	Ph Ysical
PS-EHWSN	A P reamble S ampling S cheme MAC Protocol for E nergy H arvesting WSN
RCSF	R éseau de C apteurs S ans F il
RCSF-RE	R éseau de C apteurs S ans F il à R écupération d' E nergie
RIMAC	R eceiver I nitiated MAC
RTS/CTS	R equest T o S end/ C lear T o S end
S-MAC	S ensor- MAC
SCP-MAC	S cheduled C hannel P olling- MAC
SNIR	S ignal to N oise and I nterference R atio
T-MAC	T imeout- MAC
TCP	T ransmission C ontrol P rotocol
TDMA	T ime D ivision M ultiple A ccess
TIC	T echnologies of I nformation and C ommunication
OS	O perating S ystem
TRAMA	T Rafic A daptive M edium A ccess
UDP	U ser D atagram P rotocol
WSN	W ireless S ensor N etwork
ZMAC	Z ebra MAC

Introduction générale

1 Contexte et motivations

Le développement technologique actuel dans le domaine de l'électronique, mécanique et aussi le domaine de la communication ont permis le développement de nouveaux dispositifs à faible coût transformant d'une grandeur physique détectée dans l'environnement, en une grandeur exploitable dans beaucoup de domaines préventifs, nommés capteurs. Ces capteurs capturent des informations de leur environnement, les traitent (ou pas) et les communiquent à l'utilisateur final. La coopération d'un grand nombre de ces capteurs et d'une station de base (appelée le « *Sink* » ou le « *Getway* »), donne naissance à un réseau de capteurs sans fil. Les réseaux de capteurs sans fil (*RCSF*) sont des réseaux sans infrastructure et qui sont devenus de plus en plus importants du fait qu'ils sont présents dans de nombreuses applications, telles que l'automatisation de processus industriels, les systèmes de contrôle aérien et le contrôle des patients dans des hôpitaux, etc. [1] [2] [3]

Ces réseaux sont généralement caractérisés par un grand déploiement, à une grande échelle dans des environnements où les ressources sont limitées, telles que la capacité de traitement, de stockage et surtout l'énergie car ils sont régulièrement alimentés par des batteries limitées puisque celle-ci, une fois consommée, doit être changée ou rechargée. D'autre part, ces types de réseaux peuvent être utilisés dans des emplacements isolés ou hostiles où il est difficile d'enlever les batteries pour les remplacer (des forêts, ou des montagnes difficiles d'accès ou des rivières, etc.) ou les recharger tels que les réseaux de zone corporelle (*BAN : Body Area Networks*) [4]. Du fait que chaque capteur possède une quantité d'énergie limitée, il joue à la fois, le rôle d'hôte et le rôle de routeur. Par conséquent, il utilise sa propre énergie pour acheminer des paquets destinés à d'autres nœuds du réseau. Bien que les performances du *RCSF* dépendent de plusieurs critères tels que la qualité de service [5] [6] [7], sans aucun doute la gestion de la consommation d'énergie des capteurs joue un rôle important dans la durée de vie du réseau qui est devenu le critère de performance prédominant dans ce domaine. Il devient alors important d'introduire de nouveaux schémas permettant d'administrer la consommation de l'énergie de manière efficace.

Pour améliorer les performances des *RCSF* et leur durée de vie, plusieurs travaux de recherche existants ont pour but d'économiser la consommation énergétique des capteurs à travers l'utilisation de schémas de conservation innovants, spécialement l'augmentation de sa durée de vie tels que le regroupement (le *clustering*) et l'équilibrage de charge (*load balancing*) [8] [9] [10] [11] [12] et [13]. Parmi ces schémas, il y en a un dénommé « *Duty-cycling* », qui nous paraît fort intéressant, [14] où chaque nœud dans le réseau

peut basculer entre l'état de réveil et l'état de veille, ce mécanisme est assuré par le protocole *MAC* (*Medium Access Control*), et peut être divisé en deux catégories: synchrones et asynchrones. Dans les protocoles *MAC* synchrones, chaque nœud est synchronisé avec son voisinage où chaque nœud émetteur peut transmettre un paquet à un récepteur prévu pendant leur période d'écoute sur la base de l'utilisation des messages de contrôles. L'objectif essentiel de ces protocoles est de réduire les pertes d'énergie. Par contre, les protocoles asynchrones réduisent la consommation d'énergie sans l'utilisation des messages de contrôle, les nœuds non pas besoin de synchronisation, un nœud émetteur peut attendre longtemps avant de transmettre le paquet.

Cependant, la durée de vie reste toujours limitée et bornée. Une autre alternative a récemment émergé et consiste à intégrer un système de récupération d'énergie [15] [16] dans laquelle chaque nœud capteur afin d'obtenir son énergie vitale, utilise l'environnement externe (lumière solaire et artificielle, mouvements vibratoires, chaleur, etc.) et de convertir cette énergie en une énergie électrique stockée ou directement utilisée par les nœuds capteurs dans le but d'augmenter la durée de vie et pourquoi pas d'avoir un réseau à une durée de vie infinie.

Par conséquent, afin d'administrer la consommation d'énergie et la manière dont elle est récoltée par les nœuds dans les *RCSF*, la majeure partie du duty-cycle est dépensée dans un état de veille de faible consommation où les nœuds peuvent récolter plus d'énergie, ce qui entraîne une diminution du temps d'activité des nœuds, réduisant ainsi leur duty-cycle [17]. Dans le cas où un transfert multi-saut pour un paquet est souhaité, une activité de voisinage élevée entraîne une augmentation de la latence et conduit également à des collisions de paquets lorsqu'il existe plusieurs transmissions de paquets. Par conséquent, la transmission est reportée par les expéditeurs concurrents. Étant donné que la plupart des sources d'énergie ambiantes ont des caractéristiques diverses qui dépendent des différentes situations environnementales, l'objectif principal, alors, est de réduire le duty-cycle des nœuds en fonction de leurs niveaux d'énergie afin d'atteindre un équilibre de charge et d'énergie entre les nœuds en améliorant ainsi les performances du *RCSF*.

C'est dans ce contexte que se situe notre travail de thèse de doctorat, nous nous sommes donc intéressés à l'utilisation de la technique de la récupération d'énergie dans les *RCSF* pour la proposition de nouvelles solutions de gestion de l'énergie dans ce type de réseau. Qui tiennent compte de facteurs particuliers dans lesquels peuvent évoluer ces réseaux à savoir la planification des périodes de veille et de réveil d'un capteur sur la base de son niveau d'énergie restant dans sa batterie et afin qu'il puisse récolter plus d'énergie. En présence de cette planification, plusieurs solutions de gestion de l'énergie peuvent être recommandées.

2 Objectifs de la thèse

L'objectif de ce travail de thèse consiste principalement à adapter le fonctionnement du réseau aux cycles de recharge décharge des capteurs. En effet, puisque le réveil d'un

capteur ne peut pas être estimé avec précision, du fait que le taux exact d'énergie récolté par le capteur fluctue dans le temps, il est très difficile de s'assurer de l'acheminement des paquets. De plus, l'incertitude sur le temps qu'il faut à un nœud afin de récolter assez d'énergie pour pouvoir fonctionner à nouveau, a fait en sorte que les solutions de planification de veille-réveil d'un capteur proposées dans la littérature sont inutilisables. De ce fait, l'objectif de cette thèse est de proposer une nouvelle solution de veille-réveil adaptée à la fréquence de recharge de chaque capteur, ainsi que de son importance dans le processus d'acheminement des paquets.

3 Contributions

Dans le cadre de cette thèse, nous avons exploité l'utilité de la technique de récupération d'énergie dans les *RCSF* qui a permis aux nœuds capteurs d'intégrer des systèmes de récolte d'énergie afin d'améliorer leurs performances (taux de livraison de paquets, la latence, etc.). Alors, le souci principal de notre travail est d'améliorer la performance du nœud capteur afin d'augmenter sa durée de vie tout en adaptant le fonctionnement du réseau aux cycles de recharge/décharge des capteurs. Ainsi, nos contributions de cette thèse de doctorat peuvent être subdivisées en cinq contributions décrites comme suit :

Contribution 1: Comparaison en termes d'économie d'énergie entre trois protocoles MAC asynchrones de base pour les *RCSF*

Dans cette contribution, nous avons comparé trois protocoles *MAC* asynchrones à économie d'énergie, qui conviennent intuitivement aux réseaux de capteurs sans fil à récupération d'énergie (*RCSF-RE*), car ces protocoles peuvent réellement prendre en charge les duty-cycles individuellement, mieux que des protocoles synchrones, ce qui est considéré comme une condition essentielle pour atteindre nos futurs objectifs. Nous avons aussi pris en compte l'adaptation du fonctionnement du *RCSF* au cycle de stockage et d'utilisation de l'énergie par les nœuds de capteur; deux protocoles reposent sur l'écoute à faible consommation (*LPL*) (*BMAC* [18] et *XMAC* [19]) et l'autre sur le paradigme de balisage (*RIMAC* [20]). Grâce à des simulations approfondies utilisant *OMNeT++* / *MiXiM* [21] [Web 01], nous fournissons une comparaison montrant que *RIMAC* est le plus économe en énergie que *BMAC* et *XMAC*.

Contribution 2: L'implémentation d'un module de la récolte énergétique pour les *RCSF* dans le modèle de simulation *MiXiM* sous le simulateur *OMNeT++*

Afin de simuler nos contributions 3, 4 et 5, le besoin d'implémenter un nouveau module de récupération d'énergie dans le modèle de simulation *MiXiM* sous le simulateur *OMNeT++*, s'est fait sentir pour plus d'objectivité.

Contribution 3: La proposition d'un mécanisme de gestion de l'énergie pour les *RCSF* à récupération d'énergie.

Pour améliorer les performances du *RCSF* et sa durée de vie, la gestion de la consommation d'énergie et de la récupération d'énergie est cruciale. Dans cette

contribution, nous avons proposé un mécanisme de gestion de la consommation d'énergie (*EH2M*) pour *RCSF*, afin de minimiser le duty-cycle sur la base de la quantité d'énergie restante. Avec *EH2M*, chaque nœud de capteur peut ajuster dynamiquement son duty-cycle en calculant la durée de sa période de veille sur la base de l'énergie restante. De plus, *EH2M* proposé fournit un équilibre entre la consommation d'énergie et la capacité de récupération d'énergie de chaque nœud de capteur dans *RCSF*. Via les simulations sous *OMNeT++ / MiXiM*, nous avons prouvé que notre mécanisme proposé améliore les performances du *RCSF* par comparaison avec d'autres mécanismes de gestion de la récupération d'énergie tels que l'état *ENO* [22] (*Energy Neutral Operation*). L'idée fut d'intégrer ces deux mécanismes dans un type de communication *MAC* initié par le récepteur.

Contribution 4: La proposition d'un système de gestion de l'énergie amélioré pour les *RCSF* à récupération d'énergie.

Dans cette contribution, nous avons proposé un système de gestion de l'énergie amélioré pour les *RCSF* appelé (*EEM-EHWSN*) en se basant sur la récolte énergétique et le mécanisme réalisé dans la contribution numéro trois. A la solution 3, nous avons ajouté un nouveau mécanisme qui estime la durée de temps de veille nécessaire pour recharger les unités de stockage d'énergie de sorte que les tâches puissent être exécutées. Ce dernier fonctionne grâce à l'inclusion d'une politique de seuil d'énergie et l'utilisation de l'énergie résiduelle d'un nœud capteur, les périodes de réveil et de veille sont ajustées afin de minimiser le duty-cycle de chaque capteur dans le réseau tout en garantissant un équilibre entre la consommation d'énergie et la capacité de récupération d'énergie. La politique de seuil d'énergie proposée a deux phases: une phase de commutation entre trois états possibles pour chaque nœud capteur pour réguler sa période de réveil et d'assurer un équilibre de charge dans le réseau et une deuxième phase pour le calcul de la durée de la période de veille afin d'optimiser le duty-cycle.

Contribution 5: La proposition d'un protocole *MAC* basé sur un schéma d'échantillonnage de préambule pour les *RCSF* à récupération d'énergie.

Bien que l'idée de déplacer l'initiation de communication de l'émetteur vers le récepteur soit intéressante, car les différents schémas utilisés résoudront certains problèmes existants dans le réseau tels que *l'overhearing*, la collision, etc., cependant de tels protocoles ne peuvent pas assurer la communication dans les *RCSF-RE* dans le cas où il n'y a pas suffisamment de nœuds de transfert éveillés. Soit parce que la densité des nœuds en service est trop basse ou encore que la période de récolte moyenne d'énergie est trop longue. Pour cette raison et en prélude d'inventer un autre protocole du type *DTN* (*Delay Tolerant Network*), nous avons opté pour la proposition d'un nouveau protocole *MAC* basé sur un schéma d'échantillonnage de préambule pour les *RCSF* à récupération d'énergie appelé (*PS-EHWSN*), afin d'établir des communications initiées par les émetteurs dans le *RCSF-RE*. *PS-EHWSN* exploite l'avantage des schémas initiés par un émetteur et utilise la technique d'écoute à faible consommation d'énergie (*LPL*) avec des messages de préambule courts permettant à chaque nœud de capteur du réseau de

déterminer sa prochaine période de veille sur la base de l'utilisation du (*EH2M*) mécanisme de gestion de l'énergie proposé dans la contribution 3 afin de réduire le duty-cycle. De plus, *PS-EHWSN* favorise une consommation d'énergie uniforme des nœuds de capteurs, selon la capacité de l'énergie récoltée.

4 Organisation de la thèse

Ce manuscrit est structuré en deux parties, la première consiste à présenter un état de l'art sur le domaine de recherche ainsi que le contexte de travail, tandis que la deuxième est consacrée à la présentation détaillée de nos différentes contributions attendue de ce travail. Il y a six chapitres organisés comme suit. Après cette introduction générale, nous consacrons le **chapitre 1** à un état de l'art sur les réseaux de capteurs sans fil. Nous donnons tout d'abord la définition d'un réseau *ad hoc*, puis nous présentons les *RCSF*, en commençant par la description des divers éléments constitutifs d'un nœud capteur suivi d'une présentation de l'architecture générale d'un *RCSF* et leurs types de base. Ensuite nous exposons premièrement, les besoins des applications et quelques domaines d'applications des *RCSF* et deuxièmement nous décrivons les différentes couches de la pile protocolaire utilisée par un nœud capteur et leurs différentes sources de consommation d'énergie dans ce type de réseau. Enfin, quelques définitions de base sont données (la couverture, la connectivité dans un *RCSF* et leur durée de vie).

Le **chapitre 2**, présente un état de l'art sur les protocoles *MAC* existants proposés pour les *RCSF* qui sont classifiés en quatre grandes catégories suivantes: protocoles basés sur la contention, basés sur la planification, basés sur l'interrogation de canal et hybrides. Nous présentons le fonctionnement détaillé, les points forts et les points faibles de plusieurs protocoles *MAC* bien connus entrant dans chaque catégorie. Enfin, nous comparons tous les protocoles examinés, en termes d'efficacité énergétique, de latence, de débit, de synchronisation et de scalabilité sous forme de tableau. Nous concluons que la plupart des protocoles *MAC* proposés sont sous-optimaux pour un paramètre donné. Ils prennent partiellement en charge un paramètre ou un compromis avec un autre paramètre, notamment en faveur de l'énergie. Ce qui motive le besoin d'une énergie renouvelable ainsi que d'un protocole *MAC* efficace pour les *RCSF*.

Ensuite le **chapitre 3**, expose un état de l'art sur les protocoles *MAC* récemment proposés pour les *RCSF-RE*. Tout d'abord nous avons fait un survol sur la notion de la récupération d'énergie à partir des environnements externes en donnant premièrement quelques définitions de base (la récupération d'énergie, des architectures des systèmes de récupération d'énergie qui peuvent être utilisées par un nœud capteur, les différentes sources externes d'énergie pour la récupération d'énergie, etc.) puis nous avons présenté quelques applications de *RCSF-RE* et les différentes motivations pour les protocoles *MAC* spéciaux pour les *RCSF-RE*. Ensuite nous consacrons le reste du chapitre à la description de plusieurs protocoles *MAC* fournis pour les *RCSF-RE* et classons ces protocoles selon le mécanisme de contention du canal utilisé par les nœuds capteurs dans

le réseau en deux catégories suivantes: les premiers utilisent le mécanisme d'interrogation probabiliste tandis que les seconds utilisent le protocole *CSMA/CA* (*Carrier Sence Multiple Access with Collision Avoidance*) et par la suite nous exposons leur fonctionnement détaillé. À la fin, une analyse théorique est faite où les avantages et les inconvénients des protocoles discutés sont cités, nous arrivons à un tableau comparatif résumant ces dits protocoles.

Dans le **chapitre 4**, nous comparons deux types différents de protocoles *MAC* asynchrones pour les *RCSF*. Le premier basé sur l'écoute à faible consommation (*LPL*) utilisant la transmission de la trame du préambule tels que (*BMAC* et *XMAC*), et le second sur le paradigme de balisage utilisant la transmission de la balise *Beacon* tel que (*RIMAC*). Nos travaux sont basés sur le simulateur de réseau *OMNeT ++/MiXiM* et les détails de l'implémentation sont fournis dans ce chapitre où les performances de conservation d'énergie sont évaluées. Le reste de chapitre est organisé comme suit : premièrement, nous allons présenter le détail du modèle de consommation d'énergie de ces protocoles, la simulation et les paramètres utilisés. Ensuite, les résultats des simulations sont discutés.

Ensuite, au **chapitre 5**, nous abordons de nouveau le problème de la gestion et de la récupération de l'énergie dans les *RCSF* afin d'améliorer les performances de ce dernier et sa durée de vie par la proposition d'un schéma de gestion de l'énergie amélioré pour les réseaux de capteurs à récupération d'énergie nommé (*EEM-EHWSN*). Dans ce chapitre nous commençons par la description de l'architecture du nœud de capteur utilisée, le schéma de communication fondamental adopté ainsi que le modèle de la consommation d'énergie, et de la latence définis dans *EEM-EHWSN*. Ensuite, nous détaillons la politique de seuil d'énergie proposée dans *EEM-EHWSN*, via les simulations faites, ce qui nous conduit à l'évaluation des performances du *EEM-EHWSN*.

Dans le **chapitre 6**, nous détaillons notre protocole *PS-EHWSN* (*A Preamble Sampling Scheme MAC Protocol for Energy Harvesting WSN*) proposé, un protocole *MAC* basé sur un schéma d'échantillonnage de préambule pour les *RCSF-RE* dont la diffusion des préambules courts est exécutée et exploite le mécanisme *EH2M* de la troisième contribution afin de minimiser le duty-cycle de chaque nœud de capteur dans le *RCSF-RE*. Dans ce chapitre, nous présentons en premier lieu l'architecture du nœud de capteur utilisée, les opérations utilisées dans le protocole *PS-EHWSN* ainsi que le modèle de la consommation d'énergie et de la latence sont introduits. En deuxième lieu, nous décrivons les outils de simulation et les paramètres utilisés et par la suite l'évaluation des performances du *PS-EHWSN* est donnée.

Enfin, cette thèse se termine par une conclusion générale sur les travaux de recherche présentés et identifie la contribution principale qui met en évidence les avantages du système proposé. Elle indique également les perspectives pour de futurs travaux.

PARTIE I
ÉTAT DE L'ART ET
CONTEXTE

Les Réseaux de Capteurs Sans Fil

1.1 Introduction

Les réseaux de capteurs sans fil (*RCSF*) sont devenus d'une grande importance et constituent une nouvelle étape dans le développement des technologies de l'information et de la communication (*TIC*). Cette nouvelle thématique attire de plus en plus l'attention en raison de la diversité de ses applications dans de nombreux domaines et en particulier dans l'observation des conditions physiques ou environnementales. Le souci le plus important dans les *RCSF* est comment réduire la consommation d'énergie utilisée par les capteurs qui sont habituellement alimentés par des piles ou des batteries disposant d'une capacité de stockage bornée où leur manipulation manuellement est difficile lors de leur remplacement ou leur rechargement. Afin d'améliorer la durée de vie du *RCSF* plusieurs recherches ont été proposées, qui ont eu pour idée d'utiliser des techniques de conservation d'énergie afin de remédier aux problèmes comme la durée de vie limitée des nœuds dans un *RCSF*. Dans ce chapitre, nous définissons d'abord les réseaux *ad hoc* qui constituent la catégorie parente des *RCSF*. Ensuite nous exposons l'architecture du capteur, l'architecture générale du *RCSF* et leurs types de base, quelques domaines d'applications des *RCSF*, les différentes couches de la pile protocolaire utilisée par un capteur et leurs différentes sources de consommation d'énergie dans le *RCSF*. Enfin, nous donnons quelques définitions telles que la couverture, la connectivité et la durée de vie d'un *RCSF*.

1.2 Les réseaux *ad hoc*

Un réseau mobile *Ad Hoc* est un réseau sans fil, appelé généralement *MANET* (*Mobile ad hoc network* en anglais) ou appelé aussi un *réseau multi-saut sans fil* (*wireless multihop network*), comprend une grande masse relativement dense de stations mobiles (généralement appelées hôtes ou nœuds) qui se déplacent librement dans un territoire quelconque et qui peuvent communiquer au moyen des « *ondes radio* » qui se propagent entre les différentes stations mobiles sans existence d'aucune infrastructure préétablie ou une administration centralisée [23]. Ainsi, ces réseaux sont caractérisés par l'auto-configuration et l'auto-organisation. Puisque les nœuds source et destination peuvent ne pas être dans une même portée radio dans ce type de réseau, alors la communication entre les différents nœuds mobiles est faite en deux modes, le premier, c'est le mode direct (*point-a-point*) où un nœud peut communiquer directement avec un autre nœud dans le réseau s'il est situé dans sa zone de transmission, par contre le

deuxième est le mode indirect (*multi-sauts*) où un nœud peut communiquer indirectement avec un autre situé en dehors de sa zone de transmission à travers plusieurs nœuds intermédiaires dans le réseau [24] [25]. De ce fait, chaque nœud dans ce type de réseau peut jouer le rôle d'un émetteur, récepteur ou routeur capable d'assurer une communication en relayant le trafic réseau vers d'autres nœuds [26].

1.3 Les réseaux de capteurs sans fil

1.3.1 Architecture d'un nœud capteur

Le capteur est un dispositif d'une taille minuscule doté d'une source d'énergie limitée, permettant de détecter un phénomène dans un environnement proche, de traiter ou pas les données captées et enfin de les envoyer à un ensemble spécifié de destinations via un support de transmission sans fil [27]. Généralement, un nœud capteur est composé de quatre unités de base illustrées sur la figure 1.1: l'unité de capture, l'unité de traitement, l'unité de communication, et l'unité de puissance (batterie) et suivant son domaine d'application, il peut contenir aussi, des modules additionnels tels qu'un système de localisation (*GPS : Global Positioning System*), ou bien un système de récupération d'énergie (*cellule solaire*). On peut même trouver des capteurs, un peu plus volumineux, composés d'un système mobilisateur chargé de déplacer le capteur en cas de nécessité.

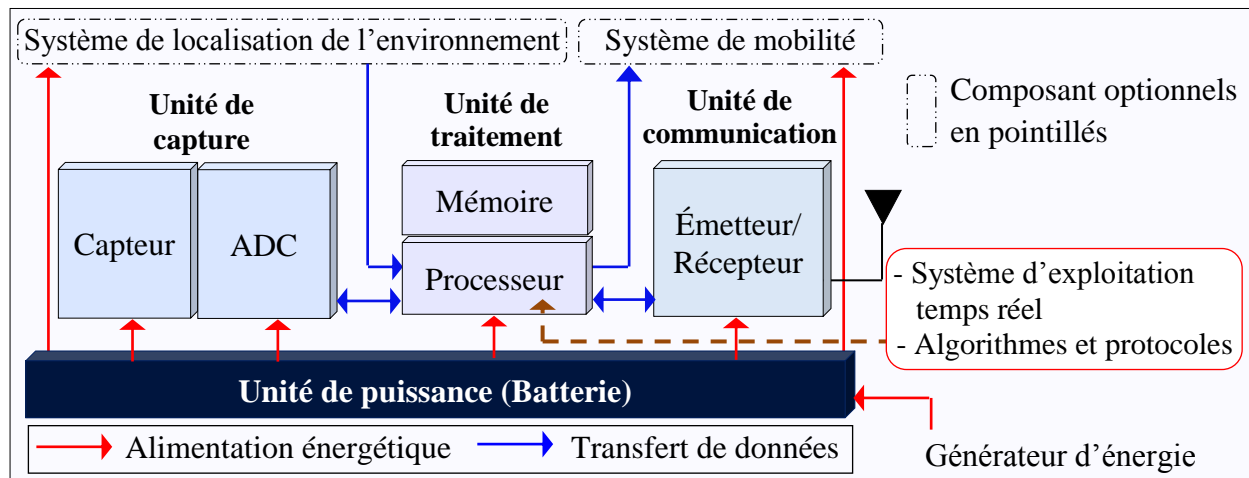


Figure 1.1: Architecture d'un nœud capteur [33]

1. L'unité de détection (*Sensing unit*)

Cette unité est généralement formée de deux sous-unités : la sous-unité de capture qui peut comprendre un ou plusieurs capteurs [28] (thermique, optique, de pression, ...) suivant le type du phénomène à détecter et la sous-unité *ADC (Analog to Digital Converter)* qui sert de convertir le signal analogique fourni par le capteur, sur la base des données rassemblées (chaleur, vibrations, humidité, ...) en un signal numérique (digital) qui va être reconnu et traité par la suite par l'unité de traitement. Vu que, il existe plusieurs types de capteurs alors le processus de sélection de l'un d'entre eux, est lié au respect de certaines métriques requises par l'application telle que la taille et la consommation d'énergie.

2. L'unité de traitement (*Processing unit*)

L'unité de traitement est considérée comme le cerveau du nœud capteur, elle comprend à la fois un processeur et une mémoire. Le processeur fonctionne à l'aide d'un système d'exploitation spécialement conçu pour les capteurs (*TinyOS* par exemple) et chargé d'exécuter des logiciels d'application et de la pile protocolaire de communications qui permettent de faire collaborer le nœud avec les autres nœuds du réseau afin de réaliser les tâches d'acquisition. Elle peut aussi analyser les données captées pour alléger la tâche du nœud puits (*sink*). Les processeurs utilisés dans les capteurs sont caractérisés par une consommation d'énergie réduite et une fréquence faible. En plus, l'unité de traitement possède deux interfaces : une interface avec l'unité de capture et une autre avec l'unité de transmission. Elle acquiert les informations en provenance de l'unité de capture et les envoie à l'unité de transmission [29].

3. L'unité de transmission (émetteur-récepteur) (*Transmission unit*)

La fonction principale de cette unité est d'assurer toutes les émissions et réceptions des données via un support de communication sans fil qui peut être de type optique, ou de type radiofréquence (*RF*) et on peut aussi l'appeler unité émetteur/récepteur radio. Elle est responsable de la modulation en émission et la démodulation en réception des données digitales sur un canal sans fil et elle contient aussi une antenne, un synthétiseur de fréquence, un oscillateur, un amplificateur et d'autres circuits nécessaires pour communiquer avec d'autres nœuds de capteurs sur le canal radio. Comme microcontrôleur, la radio peut également fonctionner dans différents modes de fonctionnement tels que la transmission, la réception, idle (mode écoute) et le mode veille. La radio est un composant important, en particulier pour les opérations d'efficacité énergétique du nœud du capteur. Cela aide à déterminer plusieurs facteurs tels que la consommation électrique, la fréquence porteuse, le débit de données, la modulation, les schémas de codage, la puissance de transmission, le blocage des erreurs et bien plus encore [30]. Les puces radios couramment utilisées sont les différentes versions de *Chipcon* (maintenant Texas Instrument) [Web 02], *MPR2400* utilisé par la série *Mica* [Web 03] et les radios *RFM TR Series* [Web 04].

4. L'unité de contrôle de l'énergie (*Energy control unit*)

Un capteur est équipé généralement d'une ressource énergétique (une batterie) pour alimenter toutes ses unités. Cependant, cette source énergétique utilisée dans ces capteurs est très limitée en raison de sa petite taille et généralement irremplaçable. La limite de cette quantité d'énergie influe directement sur la durée de vie des capteurs et par conséquent sur la durée de vie du réseau en entier. L'unité de contrôle d'énergie forme donc l'un des systèmes les plus importants. Elle est responsable de dispenser l'énergie disponible aux autres unités et de diminuer les dépenses en mettant en veille les composants inactifs par exemple. Cette unité peut aussi administrer des systèmes de récupération d'énergie ambiante (solaire, vibrationnelle, température...) à partir de l'environnement externe pour recharger les batteries des capteurs, afin de prolonger la

durée de vie du réseau. Le stockage de l'énergie se fait traditionnellement en utilisant des piles. À titre indicatif, il s'agit souvent d'une pile AA, généralement comprise entre 2.2 et 2.5 Ah fonctionnant à 1.5 V [31].

En plus de ces unités de base et conformément aux exigences de l'application, un capteur peut également être équipé d'unités supplémentaires telles que [32]:

5. L'unité de localisation (*Location finding system*)

Pour certaines applications de localisation, nécessitant la connaissance de l'emplacement physique, le capteur peut être associé à un composant *GPS* pour pouvoir déterminer leur position.

6. Le générateur de puissance (*Power generator*)

Permet de générer de l'énergie à partir de l'environnement externe.

7. L'unité de mobilité (*Mobilizer*)

Pour les applications mobiles, un moteur ou un mobilisateur peut être attaché pour déplacer les nœuds de capteurs. Elle est indispensable si le nœud capteur doit se déplacer pour exécuter une tâche et changer leur position. Dans ce qui suit, certaines des caractéristiques des plates-formes des modèles des nœuds de capteurs actuellement disponibles qui comportent les mêmes composants de base sont présentés dans le tableau 1.1 tandis que la figure 1.2 présente quelques modèles. Les modèles les plus courants sont :

1. les capteurs *MICA* [Web 13] développés par des chercheurs à l'université de Berkeley et commercialisés par *Crossbow*;
2. les capteurs *Imote* (*Imote 2* et *Tmote sky* [Web 05] [Web 06]) développés par des chercheurs d'Intel Santa Clara et commercialisés par *Crossbow*;
3. les capteurs *TinyNode* développés par la compagnie *Shoskfidh SA* [Web 07], pour les applications réelles liées à l'industrie;
4. il y a d'autres architectures au stade de prototypage dédiées à la recherche, par exemple : *BTnode* [Web 08] (Europe), *EYES* [Web 09] (Europe-Suisse), *ZigbeX mote* [Web 10] (Corée du sud), *T-Engine* (Japan), *LiveNode* (France), *BEAN* (Brazilian Energy-Efficient Architectural Node) (Bresil), *Scatterweb* [Web 11] (Allemagne).

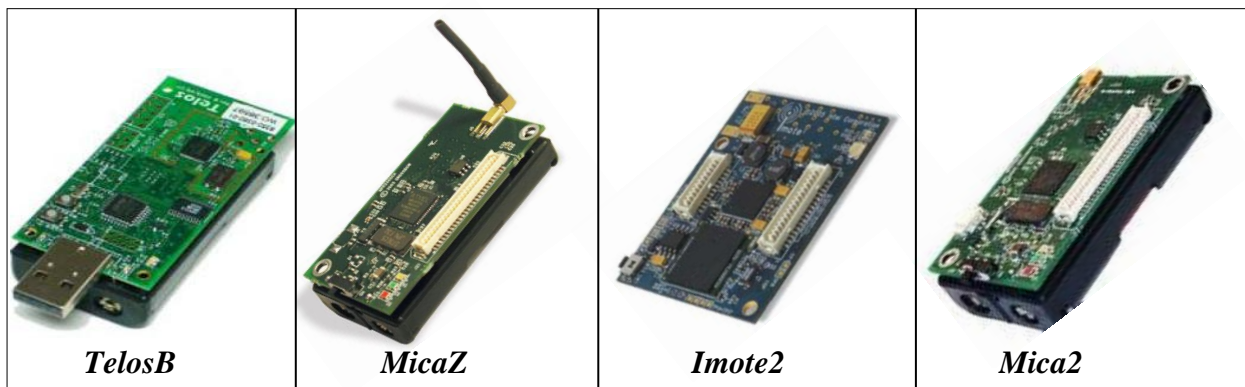


Figure 1.2: Exemples sur les modèles de capteurs sans fil [Web 13]

Capteur sans fil	Unité de traitement				Unité de transmission	Unité de captage	Unité de puissance
	Micro-contrôleur	RAM	Flash	EEProm	Type Radio		
<i>TeloSB</i> (CrossboW)	TI MSP 430 (16bits)	10 KB	48 KB	16 KB	Chipcon CC2420 250 kbps	Connecteur pour carte de capteurs externe	2xAA
<i>MICAZ</i> (CrossboW)	A Tmega 128L (8bits) 3.37 MHZ	4 KB	128 KB	4 KB	Chipcon CC2420 250 kbps	Connecteur pour carte de capteurs externe	2xAA
<i>Imote2</i> (CrossboW)	Intel PXA271 XSale (32bits) 13-416MHZ	256KB + 32MB SRAM	32 KB	32 KB	Chipcon CC2420 250 kbps	Connecteur pour carte de capteurs externe	3xAAA
<i>MICA2</i> (CrossboW)	A Tmega 128L (8bits) 3.37 MHZ	4 KB	128 KB	4 KB	Chipcon CC2420 250 kbps	Connecteur pour carte de capteurs externe	2xAA
<i>TinyNode</i> (Shockfish SA)	TI MSP 430	10 KB	48 KB	16 KB	Semtech XE 1205 153 kbps	Connecteur pour carte de capteurs externe	2/3xAA
<i>BTnode 3</i> (ETH)	ATmega 1281	64 KB	128 KB	4 KB	Chipcon CC1000/Bluetooth	Connecteur pour carte de capteurs externe	2xAA
<i>Tmote Sky</i> (Moteiv)	TI MSP 430 F1611 (16 bits) 8MHZ	10 KB	48 KB	128 KB	Chipcon CC2420	Connecteur pour carte de capteurs externe	2xAA

Tableau 1.1: Les caractéristiques de quelques modèles de capteurs sans fil [127]

1.3.2 Architecture d'un RCSF

Un réseau de capteurs sans fil est généralement formé de nombreux nœuds dispersés dans un champ de captage (appelée zone de couverture ou d'intérêt) communiquant et collaborant les uns avec les autres de manière *ad hoc* [33]. Chacun de ces nœuds a un double rôle, générer les données et les renvoyer vers une ou plusieurs stations de base appelées puits (ou « *sink* » en anglais), grâce à un routage mono-saut ou multi-saut (voir figure 1.3), dans le routage mono-saut le *sink* est directement connecté à tous les nœuds et par contre dans le routage multi-saut le *sink* n'est pas directement connecté à tous les nœuds du RCSF. Le *sink* réceptionne les données collectées par tous les nœuds capteurs répartis dans la zone de couverture, et les communique au nœud gestionnaire de tâches via internet ou tout autre réseau de communication. L'utilisateur peut alors accéder à distance à ce nœud gestionnaire de tâches pour récupérer les données recensées. Cette architecture est illustrée dans la figure 1.3.

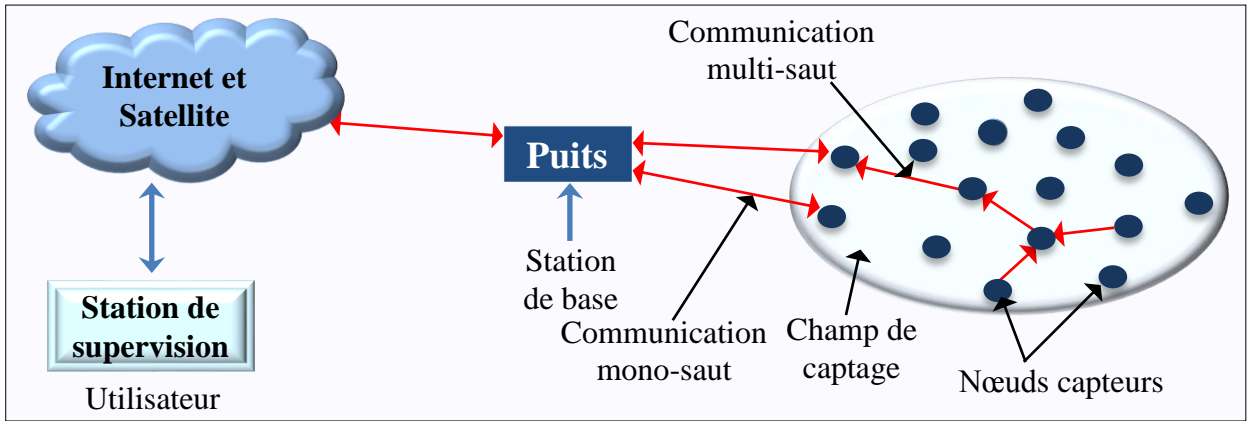


Figure 1.3: Architecture d'un RCSF [33]

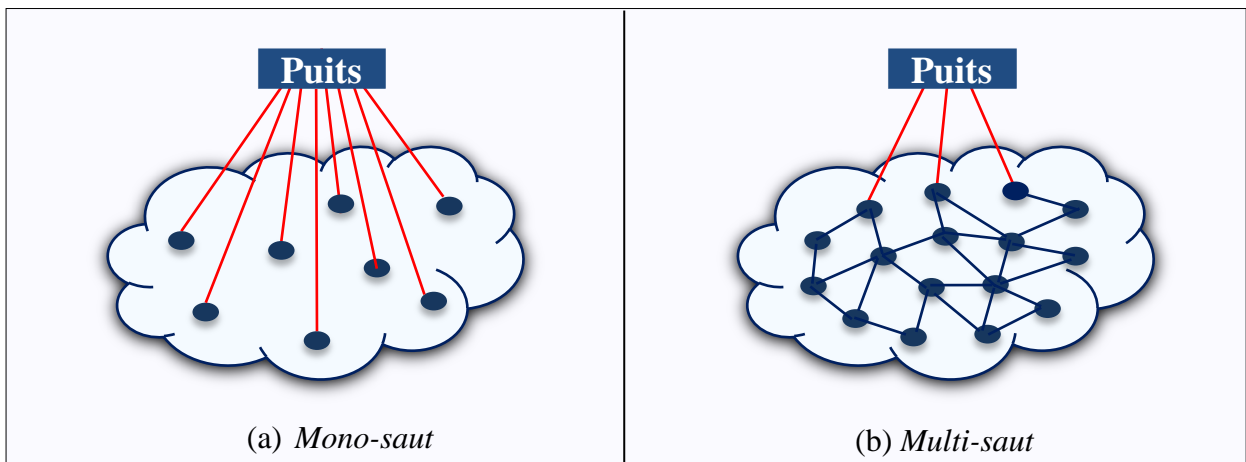


Figure 1.4: Architecture plate d'un RCSF [60]

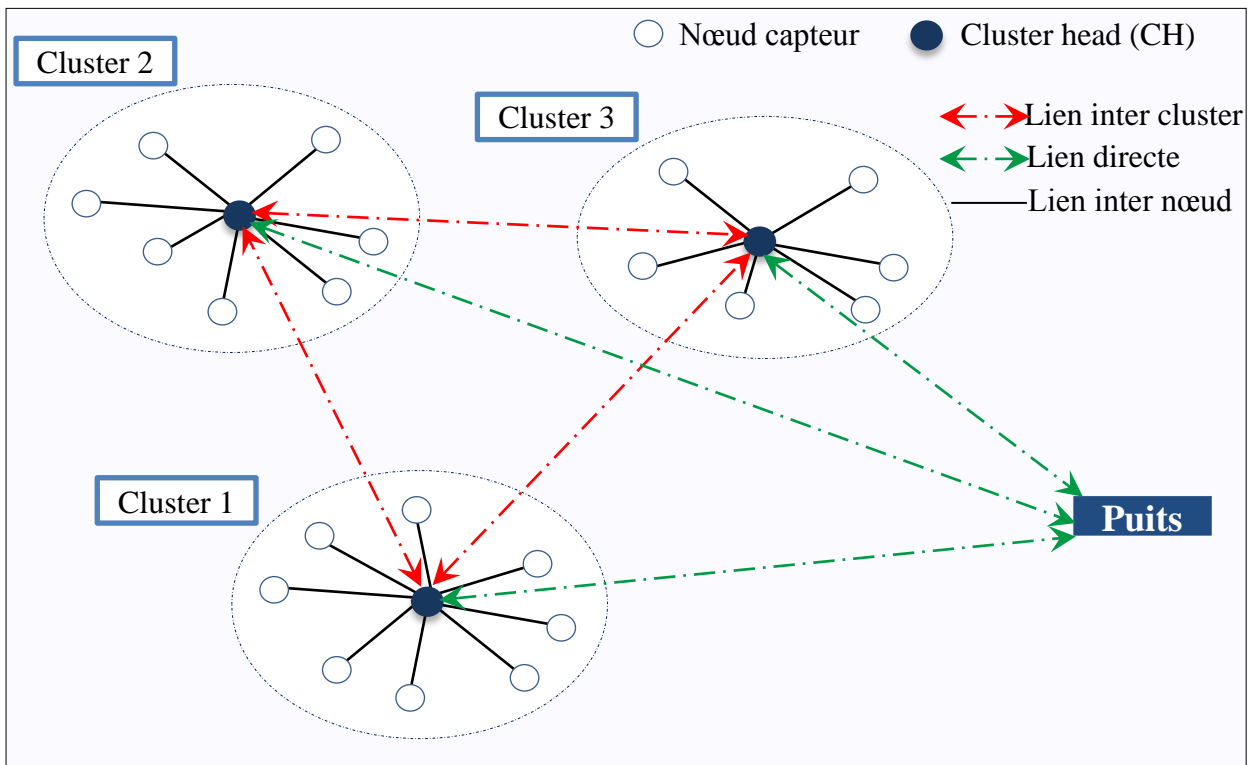


Figure 1.5: Architecture hiérarchique à base de clusters d'un RCSF [60]

L'architecture d'un *RCSF* peut être divisée en deux architectures différentes en fonction de la façon dont les nœuds capteurs communiquent entre eux [34]: la première est une architecture plate (voir figure 1.4) où chaque nœud de capteur est similaire et possède les mêmes capacités dans l'exécution d'une tâche de détection. Les nœuds capteurs forment des multiples chemins vers le nœud *sink* de manière distribuée en relayant des données à d'autres pairs. Par contre, dans l'architecture hiérarchique (voir figure 1.5), les nœuds sont arrangés en clusters, chacun étant contrôlé par une tête de cluster. Les membres du cluster envoient leurs données à la tête, qui les relaie ensuite vers le nœud *sink* d'une manière mono ou multi-saut. La tête de cluster peut avoir une capacité différente par rapport aux autres nœuds. Ces deux architectures ont leurs propres avantages et inconvénients.

1.3.3 Les types de base d'un *RCSF*

Les *RCSF* peuvent être classés en deux grandes catégories de base [35] :

1. **Un *RCSF mobile***: un *RCSF mobile* est formé d'un ensemble de nœuds capteurs mobiles (voir figure 1.6(a)) se déplaçant dans un environnement statique. Le but de tels réseaux est la plupart du temps d'explorer des zones inaccessibles ou dangereuses. Les travaux de recherche sont souvent orientés robotique, les nœuds jouant à la fois le rôle de capteur et d'actionneur.
2. **Un *RCSF stationnaire***: un *RCSF stationnaire* est formé d'un ensemble de nœuds capteurs stationnaires (voir figure 1.6(b)) utilisés pour surveiller l'apparition d'évènements sur une zone géographique. L'objectif principal de ce type de réseau est seulement la surveillance, les données mesurées sont routées en mode multi-sauts vers le *sink*. Ce dernier réceptionne les données assemblées par tous les nœuds capteurs, et les transmet au nœud administrateur de tâches à travers internet ou tout autre réseau de communication.

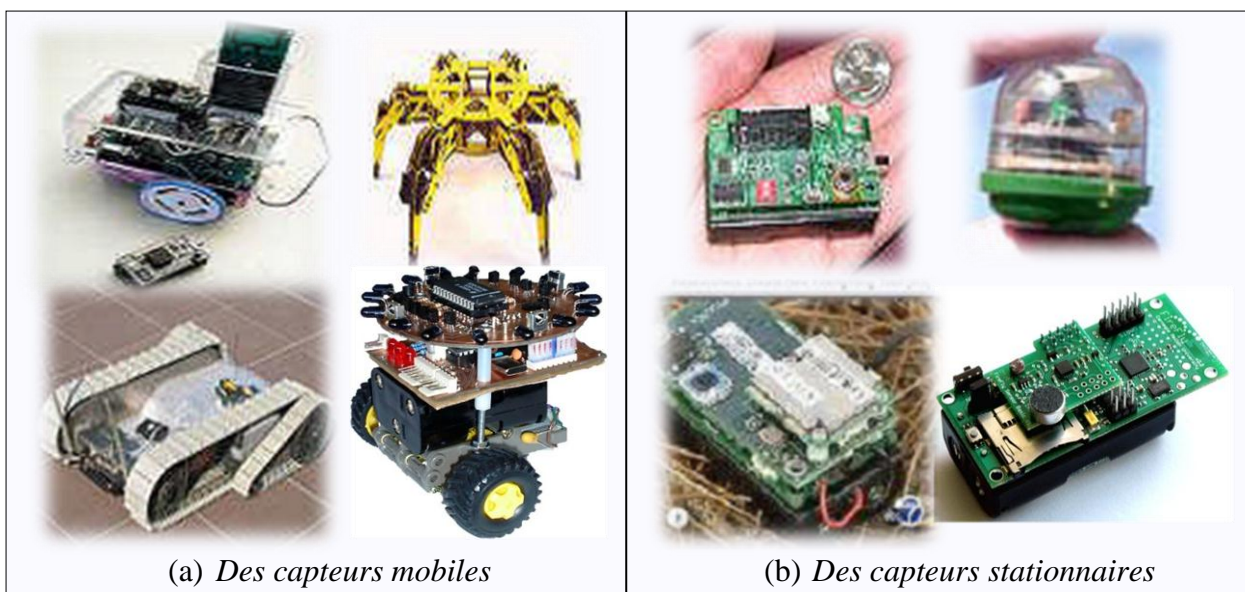


Figure 1.6: Quelques nœuds capteurs (mobiles et stationnaires) [Web 13]

1.3.4 Les besoins des applications de *RCSF*

Puisque les réseaux de capteurs sont utilisés dans de nombreux domaines et dans différents types d'applications, alors leur conception et leur mise en place sont influencées par des contraintes et des facteurs, pris également comme métriques d'évaluation des performances des protocoles dédiés à ce genre particulier de réseau. Dans ce qui suit, on présentera quelques principaux facteurs et contraintes [33].

1.3.4.1 L'efficacité énergétique

L'efficacité énergétique est une des nécessités rigoureuses lors de la conception des protocoles de *RCSF* car l'énergie est rare et trop précieuse pour en faire une problématique majeure dans ce type de réseau. En effet, les nœuds sont habituellement alimentés par une source limitée d'énergie, généralement non-rechargeable et leur remplacement est difficile voire impossible. Il devient alors important d'introduire des mécanismes permettant de gérer la consommation d'énergie de manière efficace afin de faire fonctionner les nœuds sur de longues périodes, prolongeant la durée de vie du *RCSF*.

1.3.4.2 Le passage à l'échelle (La scalabilité)

Les *RCSF* sont considérés comme des réseaux de très grandes capacités, allant de plusieurs centaines à plusieurs milliers de nœuds, répartis de manière dense où chaque nœud peut avoir un nombre important de voisins. Ainsi, une densité de réseau élevée peut entraîner des problèmes de congestion si les nœuds tentent de se communiquer simultanément, retardant ainsi la livraison des messages et la perte de paquets. Cependant, la scalabilité et l'adaptabilité sont des contraintes strictes inhérentes à la nature des applications de *RCSF*. C'est ce qui nous permet de tester l'évolutivité des protocoles de communication aux changements auxquels sont soumis les *RCSF*, en particulier les protocoles d'accès au médium *MAC* et les protocoles de routage.

1.3.4.3 La bande passante limitée

Afin de minimiser la consommation d'énergie lors du transfert des données entre les nœuds dans les *RCSF*. Les nœuds utilisent un débit de transmission faible de quelques dizaines de *Kb/s* (*kilo bit par seconde*) et qui ne représente pas un handicap pour un réseau de capteurs où les fréquences de transmission ne sont pas importantes. Par conséquent, on peut dire que les *RCSF* sont caractérisés par leurs bandes passantes limitées.

1.3.4.4 La latence

La latence fait référence à l'intervalle de temps entre l'instant où l'émetteur a un paquet à envoyer et l'instant où le récepteur reçoit ce paquet avec succès. Ainsi, elle a une grande importance qui dépend des applications des *RCSF* telles que la surveillance et

le contrôle où les nœuds sont attentifs pendant longtemps, mais il est en grande partie inactif jusqu'à la détection d'un événement. Ces applications peuvent souvent supporter une latence supplémentaire de transmission de messages. Dans une communication multi-sauts, diminuer la latence en réduisant le nombre de sauts peut permettre une économie considérable d'énergie.

1.3.4.5 Le débit

Le débit efficace fait référence à la somme de données transportées avec succès depuis un expéditeur vers une destination en un temps donné et qui mesuré souvent en bits ou octets. Ainsi, l'utilisation importante du débit engendre une grande consommation d'énergie, par contre dans d'autres applications le débit est faible ce qui permet ainsi d'économiser l'énergie. Cependant, l'utilisation du débit reste toujours liée à ses objectifs. Beaucoup de facteurs influent sur le débit, notamment la latence, l'utilisation du canal ainsi que l'efficacité des mécanismes permettant d'éviter les collisions et l'overhead.

1.3.4.6 L'équité dans le partage du canal

L'équité se réfère à l'aptitude des différents acteurs de réseau tel que les utilisateurs, les nœuds, ou les applications à partager le canal d'une façon équitable. C'est une nécessité puissante dans les réseaux classiques de voix ou de données, puisque chaque utilisateur veut avoir la même opportunité pour transmettre ou acquérir des données. Dans les réseaux de capteurs, tous les nœuds peuvent jouer des rôles distincts et coopérer pour le même objectif. De ce fait, à un instant donné, un nœud peut contenir plus de données à transmettre que d'autres. Ainsi, l'équité dans le partage du canal n'est pas une contrainte puissante pour les applications de capteurs dont la performance globale est plus dépendante de la répartition de la charge entre les nœuds que du partage du canal.

1.3.4.7 Tolérance aux pannes

La tolérance aux pannes est l'aptitude à conserver les fonctions du réseau sans interruption en cas de défaillance d'un ou plusieurs nœuds capteurs. Ainsi, la durabilité d'un nombre maximal de capteurs sans fil est une règle très importante. Dans les *RCSF*, les nœuds capteurs risquent de cesser d'exister et être en dehors du fonctionnement du réseau entier par insuffisance d'énergie et surtout lors de l'épuisement des batteries ou à cause des conditions physiques où il peut y avoir des nœuds capteurs défectueux. Ainsi, la panne de nœuds entraîne la perte des liens de communication et donc un changement significatif dans la topologie globale du réseau. Ceci peut affecter d'une façon considérable la connectivité du réseau et diminuer, en conséquence, sa durée de vie. Dans ces situations, le réseau doit être tolérant aux pannes de telle sorte que les pannes non-graves soient complètement masquées à l'application.

1.3.4.8 Auto-configuration

Parfois la topologie du *RCSF* est dynamique et dépend de la nature des applications où les nœuds sont déployés d'une manière *ad hoc*. Ainsi, le *RCSF* doit supporter des topologies étendues, denses et dynamiques. Cet aspect dynamique est dû soit à une défaillance accidentelle de certains nœuds, à l'addition ou de la perte de nœuds, à la mobilité ou la modification des modalités de dispersion. Par conséquent, les nœuds doivent avoir la capacité de s'auto-configurer au sein du *RCSF*.

1.3.4.9 Taille et coût réduit

Dans les *RCSF*, le déploiement à grande échelle des nœuds de capteurs pour les applications à contraintes spatiales, exige un coût faible et une taille réduite des nœuds. En effet, un *RCSF* doit être conçu afin que la maintenance et le support soient des plus aisés et à moindre coût. Par conséquent, les capacités de traitement et de mémorisation sont réduites. Ceci influe indirectement sur l'architecture des protocoles utilisés pour de tels réseaux.

1.3.5 Domaines d'applications des *RCSF*

Les domaines d'applications des *RCSF* augmentent de jour en jour pour plusieurs raisons, notamment: la taille miniaturisée des capteurs, leur coût de plus en plus réduit, l'existence d'une grande variété de types de capteurs (optique, vibration, thermique, ...) et surtout une grande maîtrise des supports de communication sans fil utilisés. Ces *RCSF* participent aussi dans le développement des applications actuelles et collaborent à rendre la création des autres systèmes industriels plus pratique. Parmi les domaines où ces réseaux fournissent de meilleures participations, on cite les domaines : environnemental, militaire, industriel, sécuritaire et médical. Des exemples d'applications possibles dans ces différents domaines sont énoncés dans la figure 1.7 et la figure 1.8:

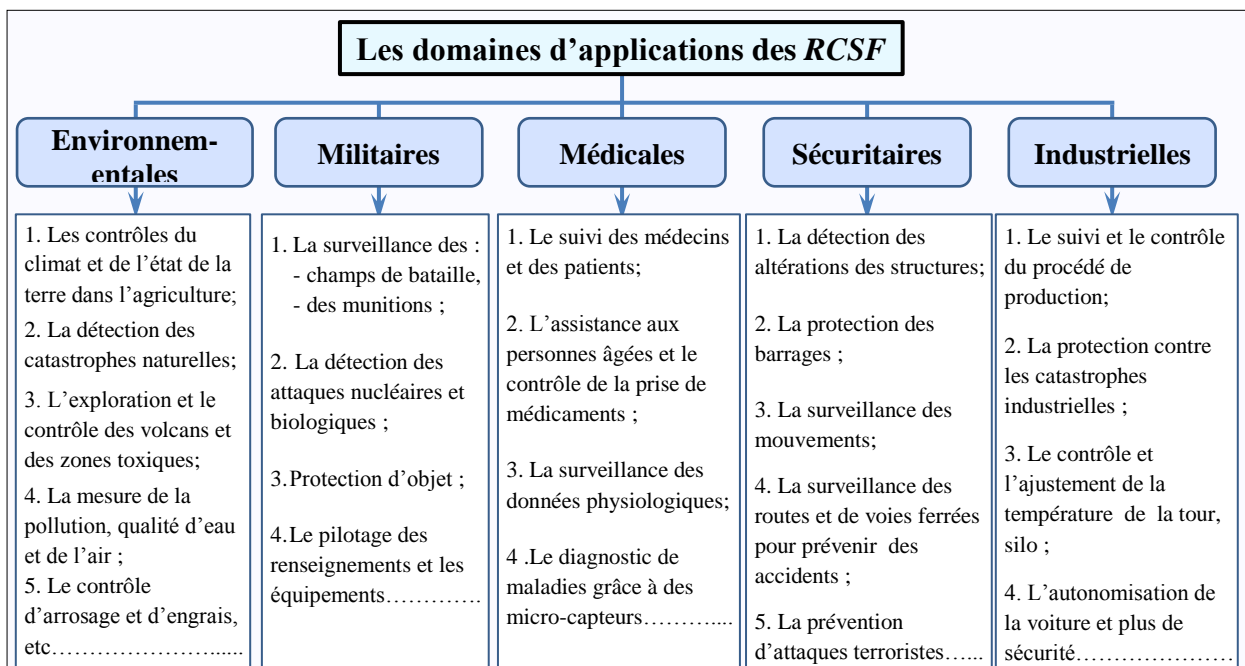


Figure 1.7: Quelques exemples des domaines d'applications des *RCSF*

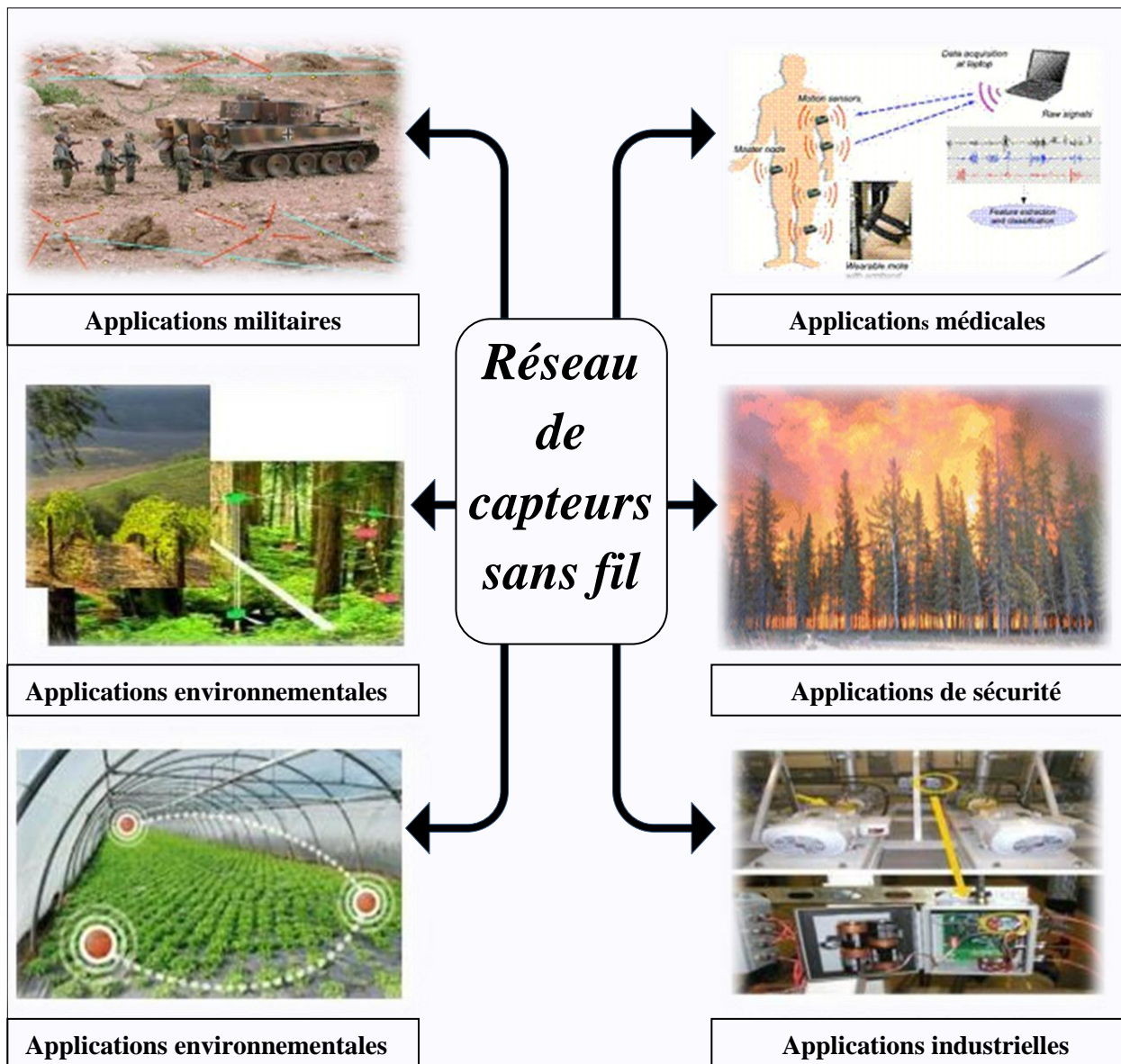


Figure 1.8: Quelques images des domaines d'applications des RCSF [Web 16]

1.3.5.1 Les applications environnementales

Grâce à la taille réduite et les capacités de plus en plus élevées en matière de calcul et de communication des capteurs, l'utilisation des *RCSF* dans les applications environnementales est devenue très utile et très importante où les capteurs peuvent être dispersés dans différents endroits difficilement accessibles afin de détecter et de surveiller des différents phénomènes qui peuvent se produire. Des exemples typiques peuvent être cités pour ces applications:

1. la capacité de détecter et de prévenir de nombreuses catastrophes naturelles telles que les inondations, la pollution et les incendies de forêt. Ce dernier est évalué par le déploiement de capteurs thermiques qui lisent périodiquement la température;
2. les contrôles du climat et de l'état de la terre dans l'agriculture permettent d'accroître l'exactitude et le rendement des activités agricoles telles que l'irrigation et l'utilisation d'engrais;

3. l'exploration et le contrôle des environnements hostiles tels que les volcans ou les zones toxiques;
4. on peut aussi utiliser les *RCSF* pour prévenir les tremblements de terre, la présence de radiations, la mesure de la qualité d'eau et de l'air, etc.

Beaucoup d'autres applications ont vu le jour, comme par exemple à l'école polytechnique Fédérale de Lausanne, où ils ont développé un nouveau système appelé *SensorScope* [Web 12] de mesures de l'environnement, distribué à grande échelle basé sur un *RCSF* ayant une capacité intégrée pour produire des mesures de densité temporelle et spatiale élevées. Ce système innovant est composé de plusieurs stations de détection à énergie solaire qui communiquent sans fil, constituant un réseau de capteurs. Les stations de détection mesuraient les principales données environnementales telles que la température et l'humidité de l'air, la température de surface, le rayonnement solaire entrant, la vitesse et la direction du vent, les précipitations, la teneur en eau du sol et son aspiration. Ces informations peuvent être très utiles pour l'humanité entière, en nous apprenant comment optimiser notre agriculture, éradiquer la faim, aider les pays souffrant de sécheresse, etc.

1.3.5.2 Les applications militaires

Comme beaucoup de techniques, le domaine militaire est le principal moteur du développement de réseaux de capteurs. Le déploiement rapide, l'auto-configuration et la tolérance aux fautes font de ce type de réseau un outil précieux dans ce domaine. Les *RCSF* dans le militaire, par exemple, peuvent être utilisés pour:

1. la surveillance des forces, des équipements et des munitions;
2. l'analyse et le contrôle du champ de bataille;
3. la détection et le traçage des objets ennemis;
4. la protection des éléments sensibles tels que les bases nucléaires et les ponts;
5. la détection et identification des attaques nucléaires, biologiques et chimiques.

Plusieurs projets ont été mis en œuvre et financés par l'Agence américaine pour les Projets de Recherche Avancée de Défense (*DARPA : Defense Advanced Research Projects Agency*). Parmi ses projets, le projet (*DSN : Distributed Sensor Network*) [36] a été l'un de ses premiers projets dans les années 1980 et le projet (*SensIT : Sensor Information Technology*) [37].

1.3.5.3 Les applications médicales

Les réseaux de capteurs dans le domaine de la médecine peuvent fournir un suivi continu des médecins et des patients, la possibilité de collecter de meilleures informations physiologiques, l'administration des médicaments et l'assistance aux personnes âgées. Cela a été rendu possible grâce à l'utilisation de micro-capteurs qui peuvent être avalés ou implantés sous la peau [38]. Parmi les principales applications dans ce domaine, on peut citer:

1. les micro-caméras pouvant être avalées et capables, sans intervention chirurgicale, de transférer des images dans le corps humain;
2. la création d'une rétine artificielle composée d'une centaine de micro-capteurs pour améliorer la vision.

1.3.5.4 Les applications liées à la sécurité

L'application des réseaux de capteurs dans le domaine de la sécurité pourrait diminuer considérablement les budgets consacrés à la sécurisation des lieux et à la protection des êtres humains tout en garantissant des résultats plus fiables. Comme exemples de ce type d'application, on peut citer :

1. la détection des altérations dans la structure d'un bâtiment, suite à un séisme ou au vieillissement, par des capteurs intégrés dans les murs ou dans le béton;
2. la protection des barrages par la détection prompte des fuites d'eau;
3. la surveillance des mouvements en constituant un système de détection d'intrusions distribué. Cet aspect va rendre plus complexe la possibilité de mettre hors d'usage ce système de surveillance.

1.3.5.5 Les applications industrielles

Grâce à leur grande flexibilité, les réseaux de capteurs sont largement utilisés dans le champ industriel. Parmi les cas d'utilisations, on cite :

1. le suivi et le contrôle du procédé de production pour assurer une meilleure qualité de service;
2. la protection contre les catastrophes industrielles en assurant un suivi efficace des équipements et produits sensibles tout en détectant précocement les problèmes;
3. l'observation d'un site susceptible de subir les effets d'une pollution et la construction en temps réel d'une cartographie de sa contamination grâce à des capteurs, disséminés sur le site, capables de relever ces niveaux pollués.

1.3.6 La pile protocolaire dans les *RCSF*

La pile de protocoles est une implémentation spéciale d'un ensemble de protocoles de communication réseau où chaque couche dépend des autres pour fournir des fonctionnalités supplémentaires. Elle suit le modèle *OSI (Open System Interconnection)* [51], sauf qu'elle doit assurer en plus la coordination entre les différents nœuds capteurs même s'ils sont en nombre de millions. En outre, elle doit respecter les caractéristiques du *RCSF*, telles que la limitation énergétique et le modèle de communication, etc. Dans le champ des *RCSF*, aucune pile de protocoles ne s'unifie. Malgré cette non-normalisation, la pile protocolaire présentée dans [33] est reprise par la majorité des documents scientifiques traitant du sujet des *RCSF*. La figure 1.9 montre la pile protocolaire utilisée par la station de base ainsi que tous les autres capteurs du réseau. Elle comprend plusieurs couches telles que : la couche physique, la couche liaison de données, la couche réseau, la couche de transport et la couche application ainsi que trois plans de gestion: le gestionnaire d'énergie, le gestionnaire de mobilité et le gestionnaire

des tâches. Selon les fonctions d'un capteur, différentes applications peuvent être utilisées et intégrées à la couche application [39].

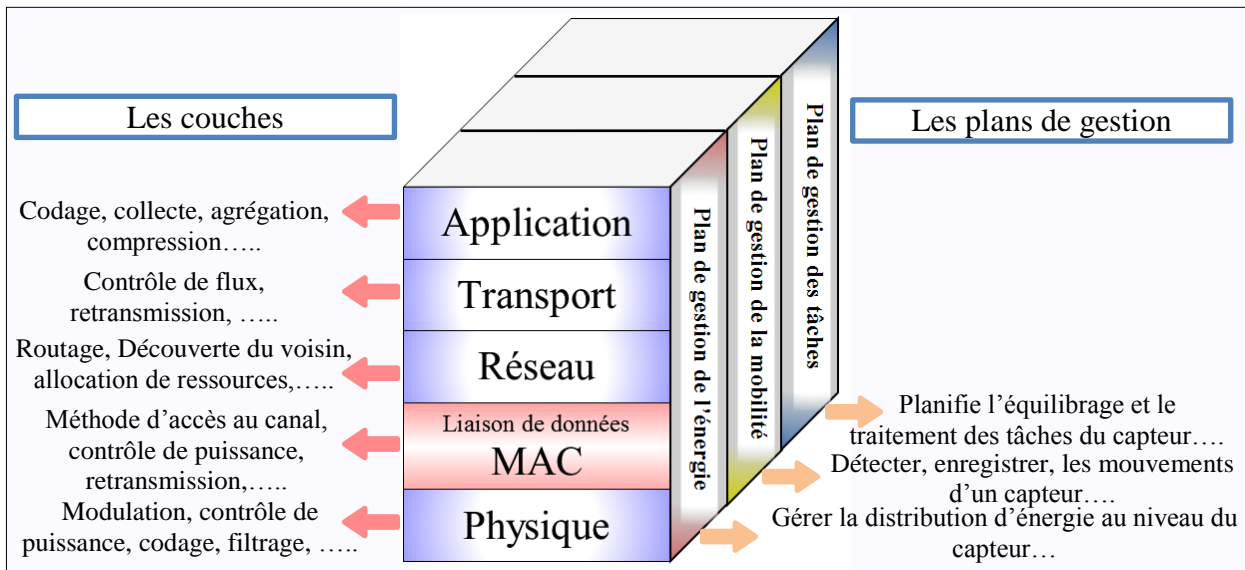


Figure 1.9: La pile protocolaire d'un RCSF [32]

1.3.6.1 La couche physique

Le rôle principal de cette couche est de garantir la connexion entre la pile protocolaire et la partie matérielle de communication réseau. En outre, cette couche permet la modulation/démodulation, le cryptage/décryptage des informations, la sélection de fréquence, la génération de la fréquence porteuse et la détection du signal. Cette dernière est assurée par l'utilisation du mécanisme *CCA* (*Clear Channel Assessment*) détaillé dans la section 2.3.2.1 du chapitre 2 pour écouter le canal physique afin de déterminer s'il est libre ou occupé pour prévenir les collisions de transmission. Au niveau de cette couche, la consommation d'énergie peut être affectée par l'environnement de l'application, le choix du type de la modulation ou la bande de fréquence utilisée. Il est intéressant en sujet d'économie d'énergie que le concepteur de la couche physique favorise une transmission à multi-sauts plutôt qu'une transmission directe qui nécessite une énergie de transmission très haute [33].

1.3.6.2 La couche liaison de données

Cette couche comprend deux sous-couches : la première nommée *LLC* (*Logical Link Control*) qui a pour rôle de détecter les erreurs des données binaires issues de la couche physique et elle permet aussi le contrôle d'erreur de transmission par l'utilisation de deux modes de contrôle d'erreur : (1) *FEC* (*Forward Error Correction*) qui représente le mode le plus utilisé dans les *RCSF* car il comprend de simples instruments de codage et de décodage [33] et (2) la retransmission *ARQ* (*Automatique Repeat reQuest*) dont leur utilité dans les applications des *RCSF* est limitée par le coût supplémentaire de retransmission et des messages de contrôle. La seconde est appelée *MAC* (*Medium Access Control*) [33]. Des détails sur la sous couche *MAC* sont exposés dans le chapitre suivant (chapitre 2 dans la section 2.2).

1.3.6.3 La couche réseau

Elle gère le routage des informations fournies par la couche transport en fournissant des chemins de communication à travers le réseau entre les nœuds sources collecteurs des données environnementales de leurs voisinages et le *sink* sur la base des protocoles de routage dédiée pour les *RCSF* en utilisant des communications multi-sauts vu les spécificités de ce type de réseau. Par conséquent, les protocoles de routage traditionnels des réseaux *ad hoc* ne conviennent pas aux *RCSF* en raison de contraintes d'énergie et de scalabilité. Parmi les métriques les plus considérées dans l'optimisation des coûts des chemins dans ce type de réseau sont : l'énergie disponible dans chaque nœud, l'énergie nécessaire pour transmettre un paquet et le temps d'acheminement des paquets. En outre, l'adressage le plus utilisé dans les *RCSF* est l'adressage géographique, c'est-à-dire que chaque nœud capteur est identifié dans le réseau par sa position géographique [40]. De plus, ce type d'adressage est particulièrement utilisé dans les applications de surveillance.

1.3.6.4 La couche transport

Elle interfère dans la communication entre deux *RCSF* ou entre un *RCSF* et Internet et plus précisément sert à maintenir le flux de données si celui-ci est imposé par le *RCSF* et notamment lors d'une connexion avec Internet. Les protocoles utilisés par cette couche doivent fournir une fiabilité diverse des paquets pour distinctes applications telles que, des mécanismes de récupération des paquets perdus et le contrôle de congestion où chaque protocole est générique et indépendant de l'application. Le *UDP* est le protocole le plus utilisé entre un nœud transmetteur et le *sink*, par contre l'emploi du *TCP* est évité vu la limite des mémoires des capteurs qui ne peuvent pas stocker des quantités de données élevées pour la gestion des communications.

1.3.6.5 La couche application

Selon la fonction des capteurs, différentes applications peuvent être utilisées et construites sur cette couche. Bien qu'il existe de nombreux domaines d'application pour les *RCSF*, les protocoles potentiels pour cette couche restent un axe majeur de la recherche. Parmi les protocoles qui ont été proposés pour cette couche dans la littérature sont [33] :

1- *SMP (Sensor Management Protocol)*: est un protocole de gestion qui fournit les opérations logicielles nécessaires pour effectuer les tâches administratives comme; l'introduction des règles liées à l'agrégation de données, à la dénomination basée sur les attributs et à la mise en cluster des nœuds de capteurs, la synchronisation temporelle des nœuds de capteurs, allumer et éteindre les nœuds du capteur, etc.

2- *SQDDP (Sensor Query and Data Dissemination Protocol)*: permet à l'utilisateur d'interroger le réseau en se basant sur la localisation des nœuds par la fourniture des interfaces pour émettre des requêtes, répondre aux requêtes et collecter les réponses

reçues par l'utilisation du langage de requêtes et le langage de commande *SQTL* (*Sensor Query and Tasking Language*).

1.3.6.6 Plan de gestion de l'énergie

Puisqu'un nœud capteur sans fil a une source d'énergie limitée, un système de gestion de l'énergie doit exister pour contrôler la façon dont chaque nœud capteur utilise son niveau d'énergie afin de maintenir une meilleure gestion d'énergie. Par exemple, avant qu'un émetteur transmette des données vers la destination, il vérifie la disponibilité du canal, si ce dernier est occupé, il se met en veille afin d'économiser la consommation d'énergie et si un nœud capteur constate que son niveau d'énergie est faible, il informe ses nœuds voisins qu'il ne peut pas participer au routage afin de préserver la quantité d'énergie restante pour la capture et la transmission.

1.3.6.7 Plan de gestion de mobilité

Permet de détecter et d'enregistrer les mouvements des nœuds capteurs et indiquer leurs positions. De cette façon, chaque nœud peut connaître ses nœuds voisins, dans le but d'assurer un équilibrage de charge et une plus juste consommation d'énergie et gestion des tâches. Il doit aussi maintenir à n'importe quel instant le chemin séparant le nœud mobile du nœud *sink*.

1.3.6.8 Plan de gestion des tâches

Permet d'assurer l'équilibrage et la répartition des tâches sur les différents nœuds de réseau relatives à une région bien déterminée tout en évitant la redondance de la collecte des données effectuées par les nœuds capteurs dans ce même endroit au même instant, alors il suffit de définir un sous-ensemble de nœuds pour garantir ces tâches suivant les niveaux de capacité énergétiques dont ils disposent et ceci dans le but d'économiser de l'énergie sur le réseau.

1.3.7 La consommation d'énergie dans les *RCSF*

Les *RCSF* souffrent de la limitation d'énergie due à la capacité limitée de la batterie des nœuds de capteurs. En effet, lorsque la batterie d'un nœud est épuisée, le nœud meurt. Lorsqu'un nombre important de nœuds meurent, le réseau peut ne plus être en mesure de réaliser la tâche qui lui est assignée. Alors, il est plus important que jamais de conserver de l'énergie pour accomplir certaines tâches. De plus, la durée de vie de ces batteries peut atteindre plusieurs mois, voire quelques années, car leur remplacement ou leur chargement est difficile surtout dans les emplacements isolés ou hostiles.

Dans tous les cas, l'énergie est une ressource importante qui devrait être optimisée régulièrement pour prolonger la durée de vie du nœud de capteur et du réseau. Dans cette section, la consommation énergétique dans les *RCSF* est analysée en mettant le point sur les formes de consommation dans un nœud capteur et les sources de pertes d'énergie.

1.3.7.1 La consommation d'énergie d'un nœud capteur

La consommation énergétique varie dans une très large plage et dépend fortement du type spécifique du nœud. Généralement les sources de consommation d'énergie sur un nœud capteur sont : le module radio utilisé pour la communication (réception/transmission), le microprocesseur pour le traitement des données, la mémoire pour la sauvegarde des données, et le capteur pour l'acquisition des données [41] [42] et [43]. La communication radio est souvent la plus consommatrice parmi les quatre. Le graphique de la figure 1.10 montre l'énergie consommée pour chaque source et pour chaque action réalisée par le nœud capteur. On voit clairement que la tâche de transmission est la plus gourmande en énergie suivie de celle de la réception, alors que la tâche de capture et celle du traitement sont négligeables.

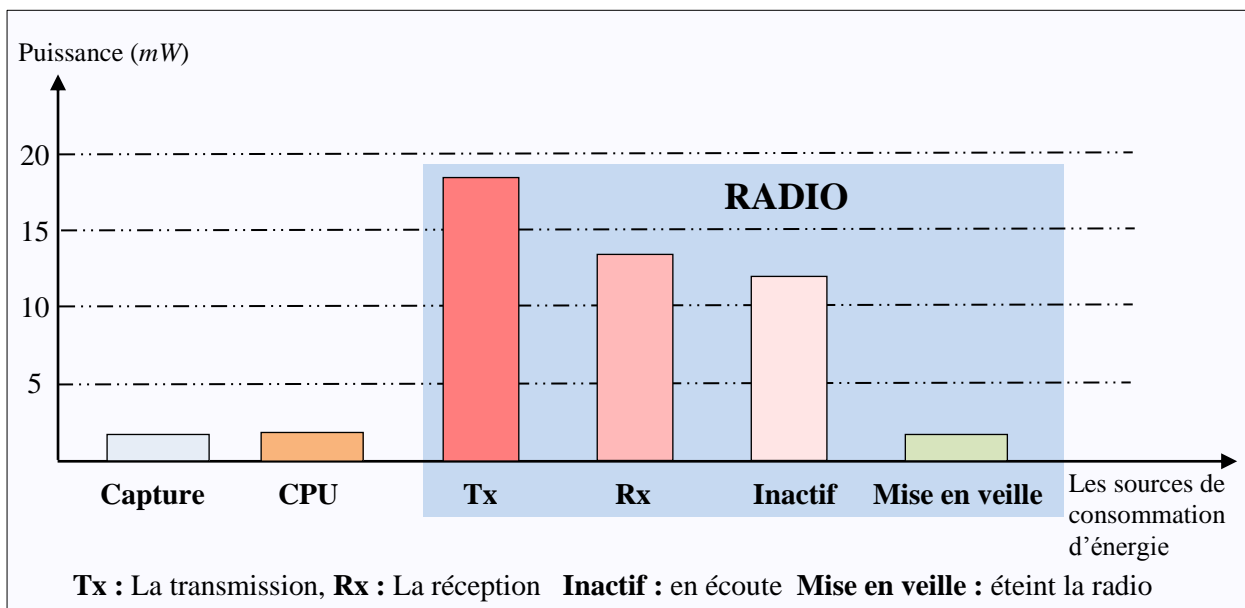


Figure 1.10: Consommation de l'énergie électrique par un nœud capteur [52]

a- La radio (réception/transmission)

Un émetteur-récepteur radio est responsable de la transmission et de la réception des paquets par voie hertzienne. En raison de la complexité de la communication sans fil, l'émetteur-récepteur doit contenir de nombreux modules fonctionnels (modules de modulation, de démodulation, de synthèse de fréquence, de conversion de fréquence, de filtrage ainsi que d'autres fonctions) et consomme une fraction importante de l'énergie totale dans un nœud de capteur. La stratégie d'économie d'énergie la plus simple consiste à éteindre autant que possible l'émetteur-récepteur et à ne l'allumer que lorsque cela est nécessaire.

b- L'acquisition des données (capture)

Il existe de nombreuses sources de consommation d'énergie par l'unité d'acquisition de données. Le coût de cette énergie dépend du type spécifique du capteur

(image, son, température, etc.) et des tâches (échantillonnage et conversion des signaux physiques en signaux électriques, conditionnement des signaux et conversion analogique-numérique, etc.) qui lui sont assignées. Ainsi, l'énergie consommée pour effectuer l'acquisition des données n'est pas très importante.

c- Énergie de traitement de données

L'énergie consommée pour les opérations de calcul est beaucoup plus faible que l'énergie de communication car les données traitées en général par les *RCSF* sont de type scalaire (température, humidité, vitesse du vent, etc.). L'énergie nécessaire pour transmettre $1kb$ sur une distance de $100\ m$ est approximativement équivalente à l'énergie nécessaire pour exécuter 3 millions d'instructions avec une vitesse de 100 millions d'instructions par seconde *MIPS* [44]. Ce niveau peut être dépassé en fonction des circuits installés dans les nœuds capteurs et des fonctionnalités requises.

1.3.7.2 Les sources de perte d'énergie

La puissance de communication du nœud dépend de plusieurs facteurs, notamment le type de schéma de modulation utilisé, le débit de données, la puissance de transmission, les modes de fonctionnement de la radio et la fréquence de commutation entre ces modes. Dans le même temps, un protocole *MAC* peut être rendu responsable des sources de gaspillage d'énergie suivantes dans les *RCSF*, qui concernent principalement la communication [45] [46].

Réception indésirable ou écoute abusive (*Overhearing*). *L'overhearing* est la réception des paquets par un nœud qui ne lui sont pas destinés. Cela est principalement dû à la nature de la transmission, qui est du type multidirectionnel qui oblige chacun des nœuds voisins à gaspiller de l'énergie et du temps de réponse sans aucun intérêt pour fournir des paquets. Ce gaspillage d'énergie devient important lorsque la charge de trafic est élevée et quand le réseau a une grande densité de nœuds.

Collisions. *La collision* est l'une des premières sources de pertes d'énergie et elle baisse aussi les performances du *RCSF*. Dans ce dernier, il peut y avoir de nombreuses interférences et de collisions dues à la nature de l'environnement de communication sans fil utilisé. La collision se produit lorsque deux nœuds transmettent des paquets simultanément et s'interfacent avec la transmission de chacun. Ainsi, il augmente la consommation d'énergie et peut également consommer de l'énergie supplémentaire par la retransmission de paquets.

L'écoute à vide (*idle listening*). *Idle listening* survient lorsqu'un nœud de capteur en mode d'écoute est prêt à recevoir les paquets entrants possibles. Cependant, le maintien de ce mode est très coûteux et inutile notamment dans les réseaux à faible trafic. Une activation périodique de la radio pour réceptionner les paquets pourrait être un moyen d'économie d'énergie.

Non disponibilité du récepteur ou envoi infructueux (L'overmitting). L'overmitting survient lorsqu'un nœud émetteur transmet des données et que le nœud destinataire n'est pas prêt à les recevoir, ce qui provoque la consommation de l'énergie de transmission. Cela impose au nœud émetteur à retransmettre ces données à nouveau afin de réussir sa transmission. Ces retransmissions consomment plus d'énergie.

Paquets de contrôle (L'overhead). L'overhead est l'utilisation de certains paquets de contrôle pour assurer le bon fonctionnement d'un protocole. Ces paquets de contrôle n'expédient pas immédiatement des données, ils diminuent également le débit utile effectif et leur échange dans le réseau affaiblit le niveau d'énergie de manière dramatique.

1.3.8 La couverture dans un RCSF

La couverture dans les RCSF est souvent considérée comme étant une mesure de performance très importante. Elle reflète la façon dont une zone donnée est surveillée (contrôlée), c'est-à-dire comment chaque point de la zone de surveillance est observé et suivi par l'ensemble des nœuds. Cette zone est souvent présentée comme un cercle de rayon r d'où le nœud à une vision qui dépend du rayon de réception de son module de communication r_s . Un nœud peut communiquer avec un autre nœud que si ce dernier se trouve dans sa zone de communication. Ainsi, on dit qu'un réseau est connecté, si les nœuds peuvent communiquer entre eux directement ou indirectement par l'intermédiaire d'autres nœuds. La figure 1.11 illustre un exemple de la zone de couverture et de communication d'un capteur.

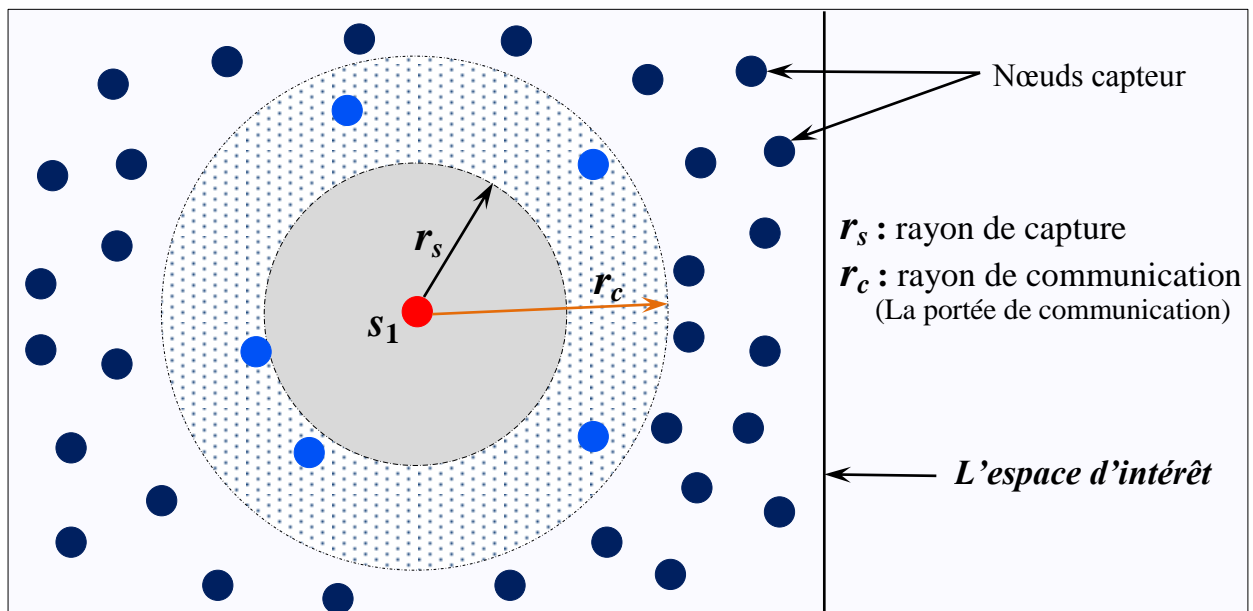


Figure 1.11: Les différents rayons (de couverture et de communication) d'un nœud capteur

La couverture d'un RCSF mesure le degré de couverture de l'espace d'intérêt par l'ensemble des nœuds capteurs. Cette couverture peut exister sous trois formes: la

couverture partielle où seulement une partie de l'espace d'intérêt est couverte par les nœuds capteurs, la couverture dense où l'espace d'intérêt est complètement couvert ou presque et la couverture redondante où plusieurs capteurs couvrent la même localisation physique. Par ailleurs, un mauvais déploiement des nœuds capteurs produira une perte de certains nœuds qui n'auront pas de voisins et qui seront isolés et déconnectés du réseau. Ainsi, une meilleure couverture est la clé pour les systèmes robustes et peut-être exploitée pour étendre significativement la durée de vie du *RCSF* en basculant les nœuds redondants en mode veille de faible consommation d'énergie.

1.3.9 La connectivité dans un *RCSF*

Un *RCSF* est dit connecté s'il y a au moins un chemin direct (connectivité à un saut) ou indirect (connectivité à multi-saut) entre chaque paire de nœuds du réseau. Dans le cas d'un réseau de ***k*-connectivité** ($k \geq 1$), il existe au moins k chemins disjoints entre toute paire de nœuds du réseau. Pour le rayon de communication r_c tel que ($r_c \geq 2r_s$) et pour une région de détection convexe, la k -couverture implique une k -connectivité [50] c'est-à-dire qu'une couverture complète garantira une connectivité complète, comme le montre la figure 1.11. Dans [50], la condition nécessaire et suffisante sous laquelle la couverture implique une connectivité où la portée de communication r_c est au moins deux fois la plage de détection r_s a été obtenue. Maintenant, si l'on considère que le rapport entre r_c et r_s est x . Alors x peut être inférieur à 1, égal à 1 ou supérieur à 1. Si la plage de transmission des capteurs est supérieure à deux fois la plage de détection, la couverture implique la connectivité. Sur la figure 1.11, nous pouvons voir que le capteur s_l peut détecter la région couverte par son rayon de détection r_s dans la région de couleur grise mais le rayon de communication r_c est au moins deux fois celui qui est indiqué par la zone avec des points.

A l'instar de la couverture, la connectivité dans les *RCSF* est considérée comme un paramètre de mesure de performance très important, surtout dans le cas des applications de *RCSF* mentionnées ci-dessus. Par conséquent, pour bien garantir toutes les fonctionnalités de telles applications, il est nécessaire de bien étudier et de prendre en compte les propriétés de connectivité lors de la conception et le déploiement de tels réseaux.

1.3.10 La durée de vie d'un *RCSF*

La conservation de l'énergie est sans aucun doute le problème le plus important dans les *RCSF* de sécurité. Comme indiqué précédemment, les nœuds de capteurs fonctionnent sur la batterie et leur recharge ou leur remplacement est souvent impossible. Cela fait de l'énergie une ressource précieuse et rare qui doit être gérée avec sagesse pour prolonger la durée de vie du *RCSF*. Il est difficile d'avoir un critère précis pour définir la durée de vie du *RCSF*. Normalement, la durée de vie de tout *RCSF* est supposée être la durée entre son déploiement et le moment où le réseau devient non fonctionnel [47].

Cependant, l'instant où un *RCSF* est déclaré comme non-fonctionnel, dépend totalement de l'usage qu'il en est fait. Ainsi, de nombreuses définitions de la durée de vie du *RCSF* existent dans la littérature, certaines d'entre elles sont indiquées de manière brève ci-dessous [48, 49]:

1. le moment où une certaine partie de nœuds meurt ;
2. le moment où le chef de la première grappe (cluster) meurt ;
3. l'heure à laquelle tous les nœuds meurent dans le réseau ;
4. l'heure à laquelle le réseau se divise en deux segments ou plus ;
5. la connectivité et la couverture du réseau sont perdues ;
6. l'heure à laquelle chacune des cibles n'est pas couverte par au moins un ou K nœuds ;
7. l'heure à laquelle le réseau n'est pas en mesure de fournir des tâches spécifiques à l'application ;
8. Etc.

1.4 Conclusion

Les *RCSF* sont devenus de plus en plus importants et envahissent notre vie quotidienne de plus en plus, du fait de leur grand champ d'application dans différents domaines et ainsi ces types des réseaux forment un appui de recherche très productif. Cependant, il reste encore de nombreux défis à relever afin de pouvoir les utiliser dans des cas réels, en toute sécurité et de manière efficace. Pour les *RCSF* la problématique majeure est la consommation d'énergie, car les capteurs sont dotés d'une capacité énergétique limitée. Dans ce chapitre, nous avons exposé d'une façon générale les *RCSF* en commençant par la définition des réseaux *ad hoc*, ensuite nous avons fait un survol sur leur architecture (du capteur et du *RCSF*), les types de base d'un *RCSF*, les besoins des applications du *RCSF*, quelques domaines d'application et la description de la pile protocolaire utilisée par un capteur. À la fin de ce chapitre, nous avons analysé la consommation d'énergie dans les *RCSF* en donnant quelques définitions sur des paramètres de mesure de performance très importante pour les *RCSF* (la couverture, la connectivité et leur durée de vie). Dans le prochain chapitre, nous nous sommes attaqués à l'état de l'art sur les protocoles de la sous couche *MAC* particulièrement proposés pour les *RCSF* en vue d'économiser l'énergie, car c'est là où se trouve notre intérêt dans cette thèse.

Les protocoles MAC et la conservation d'énergie dans les RCSF

2.1 Introduction

Dans les *RCSF*, la conservation de l'énergie est une fonctionnalité indispensable en raison de la nature des applications utilisant les *RCSF* et des périphériques à base de batteries composant ce type de réseau. Dans cette catégorie de réseau, la sous couche *MAC* est responsable des procédures d'accès aux canaux, de la planification, de l'accès à des tampons et à la gestion des supports partagés et du contrôle des erreurs, tout en offrant la fiabilité et l'efficacité. Il est également très important d'améliorer la durée de vie du réseau en gérant les communications entre les nœuds de capteurs. Cette dernière est assurée par des protocoles *MAC*. Dans ce chapitre nous présentons un état de l'art sur les différents types de protocoles de *MAC* utilisés dans les *RCSF*, tout en donnant leur classification, leur fonctionnement en détail et par la suite nous comparons ces protocoles observés. À la fin, nous présentons quelques d'autres techniques de conservation d'énergie citées dans la littérature utilisées dans les *RCSF*.

2.2 La sous couche MAC

La sous couche *MAC* est une sous-couche de la couche liaison de données spécifiée dans la pile protocolaire de communication illustrée à la figure 2.1, responsable de gérer le contrôle de l'accès au canal entre les nœuds qui veulent communiquer dans le réseau grâce à un protocole de gestion du trafic de communication, de réduire la consommation d'énergie par la désactivation de la radio d'un nœud (le mode veille) afin de prolonger la durée de vie du *RCSF*. Elle fait aussi le contrôle des activités de la radio et elle fabrique les trames de communications, via l'insertion de diverses informations (bits additionnels, entêtes, somme de contrôle...) essentielles au récepteur pour déterminer le début et la fin d'une trame ou chaque champ utile qui la compose (adresse, données, champs de synchronisation) afin de définir la mise en forme syntaxique des données issues de la couche physique, d'initier les communications entre les entités, de gérer les modes de communication ou encore de détecter les éventuelles erreurs introduites par le canal de propagation. Ces différentes phases doivent s'alterner tout en essayant de fournir un accès fiable, une faible latence et un débit équitable pour tous les nœuds. Cependant, pour que les protocoles *MAC* soient efficaces ils doivent utiliser judicieusement la radio pour conserver son énergie. Ainsi, le protocole *MAC* permet d'atteindre les objectifs de conception importants des *RCSF* en spécifiant comment les nœuds utilisent la radio,

se partagent le canal, évitent les collisions dans les environnements corrélés et de diffusion, répondent aux requêtes et survivent plus longtemps.

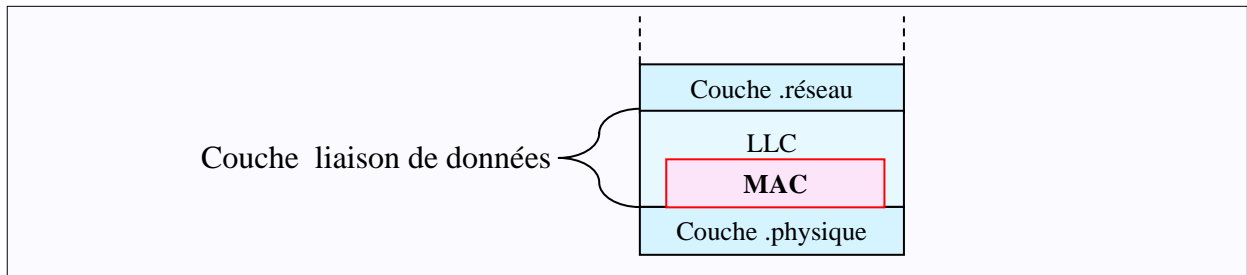


Figure 2.1: La structure de la couche liaison de données

2.3 La technique de cycle d'activité (*Duty-cycle*)

Dans les *RCSF*, le cycle d'activité d'un nœud (*duty-cycle*) est l'une des techniques de gestion de l'énergie de base et les plus couramment utilisées [52], où chaque nœud capteur peut être commuté entre deux états :

1. **Etat de veille:** le nœud éteint sa radio pour économiser plus d'énergie à chaque fois que la communication n'est pas nécessaire et lance un temporisateur. Lorsque ce dernier expire (durée de veille), il passe automatiquement à l'état de réveil.
2. **Etat de réveil:** le nœud allume sa radio pour échantillonner, calculer et communiquer pendant sa période de réveil.

En fonctionnement en *duty-cycle*, la plus grande partie du cycle est consommée dans l'état de veille de faible puissance. Cette procédure, qui dépend du support matériel [53] pour la réalisation des états de veille, permet de minimiser la consommation électrique moyenne d'un nœud grâce à de nombreux ordres de grandeur, un nœud actif ne recevant et transmettant qu'une petite partie du temps actif. Le *duty-cycle* d'un nœud i (Dc_i) est défini comme le rapport entre le temps de réveil ($t_{réveil_i}$) et la somme du temps de réveil et le temps de veille (t_{veille_i}) [17]:

$$Dc_i = \frac{t_{réveil_i}}{t_{réveil_i} + t_{veille_i}} \quad (2.1)$$

Cette forme de *duty-cycle* est administrée par des protocoles de communication de la sous couche *MAC* (*Medium Access Control*).

2.3.1 Les protocoles *Sleep/Wakeup* (*Veille/Réveil*)

Comme cité préalablement, le module de communication (la radio) d'un nœud capteur est la base de l'implémentation du régime de *veille/réveil*. Dans cette section, nous expliquerons les principaux plans de *veille/réveil* implantés en tant que protocoles indépendants du protocole *MAC*. Selon la classification présentée dans le document [55], ces protocoles sont divisés en trois grandes catégories : les protocoles à la demande, à rendez-vous programmés, à régimes synchrones.

▪ **Les protocoles à la demande.** Afin de minimiser la consommation d'énergie au niveau de chaque nœud de capteur, ces protocoles reposent sur l'idée qu'un nœud doit être réveillé juste au moment où il doit communiquer avec un autre nœud voisin. Cela, rend ce type de protocoles particulièrement appropriés pour les applications de réseau de capteurs avec un très faible duty-cycle (par exemple, détection d'incendie, surveillance de pannes de machine et, plus généralement, tous les scénarios liés aux événements). La difficulté de ces protocoles est de savoir comment avertir un nœud en veille qu'un autre nœud veut communiquer avec lui. Pour cette raison, ces systèmes nécessitent généralement deux canaux distincts: un canal de données pour la communication de données normale et un canal de réveil pour réveiller les nœuds en cas de besoin. Bien qu'il soit possible d'utiliser une seule radio avec deux canaux différents, toutes les propositions reposent sur deux radios différentes. Cela permet de ne pas différer la transmission du signal sur le canal de réveil si une transmission de paquets est en cours sur l'autre canal, réduisant ainsi la latence de réveil. L'inconvénient est le coût supplémentaire pour la deuxième radio. Cependant, ce coût supplémentaire est limité car le système radio représente généralement un faible pourcentage du coût total d'un nœud de capteur. Le protocole *STEM* (*Sparse Topology and Energy Management*) [56], par exemple, utilise deux radios.

▪ **Les protocoles à rendez-vous programmés.** Ces protocoles sollicitent que tous les nœuds voisins se réveillent en même temps. Généralement, les nœuds se réveillent périodiquement pour vérifier les communications potentielles. Ensuite, ils reviennent dormir jusqu'au prochain rendez-vous. Le principal avantage de ces systèmes est que lorsqu'un nœud est réveillé, il a la garantie que tous ses voisins sont également réveillés. Cela lui permet d'envoyer des messages de diffusion à tous les voisins [Web 14]. Spécifiquement, les nœuds se réveillent suivant un ordonnancement de réveil et restent réveillés pendant un court intervalle de temps pour communiquer avec leurs voisins. Ensuite, ils se rendorment jusqu'au prochain rendez.

▪ **Les protocoles *Sleep/Wakeup asynchrone*.** Contrairement aux protocoles à rendez-vous, ces protocoles permettent à chaque nœud de se réveiller indépendamment des autres quand il veut et tant qu'il est capable de communiquer avec ses voisins en garantissant à ces derniers, des périodes actives qui se chevauchent dans un nombre de cycles spécifié. Ce fondement est obtenu selon les propriétés incluses dans le schéma *sleep/wakeup*, donc aucun échange d'informations n'est obligatoire entre les nœuds. Dans [57] des schémas *sleep/wakeup* sont détaillés.

2.3.2 La classification des protocoles MAC de faible Duty-cycle

Plusieurs protocoles MAC pour les RCSF ont été proposés dans la littérature. La plupart d'entre eux mettent en œuvre un schéma de faible duty-cycle pour la gestion de l'énergie. Nous présentons ci-dessous les protocoles MAC les plus courants.

Afin de répondre aux exigences de conception strictes des RCSF, de nombreux protocoles MAC ont été proposés dans la littérature qui dépendent de la manière dont ils

permettent aux nœuds de capteurs d'accéder au canal. La plupart entre eux mettent en œuvre un schéma de duty-cycle faible pour la gestion de l'énergie. Ces protocoles, peuvent être classés en quatre catégories générales : des protocoles basés sur la contention, basés sur la planification, sur l'interrogation de canal et hybrides [58]. La figure 2.2 illustre cette classification.

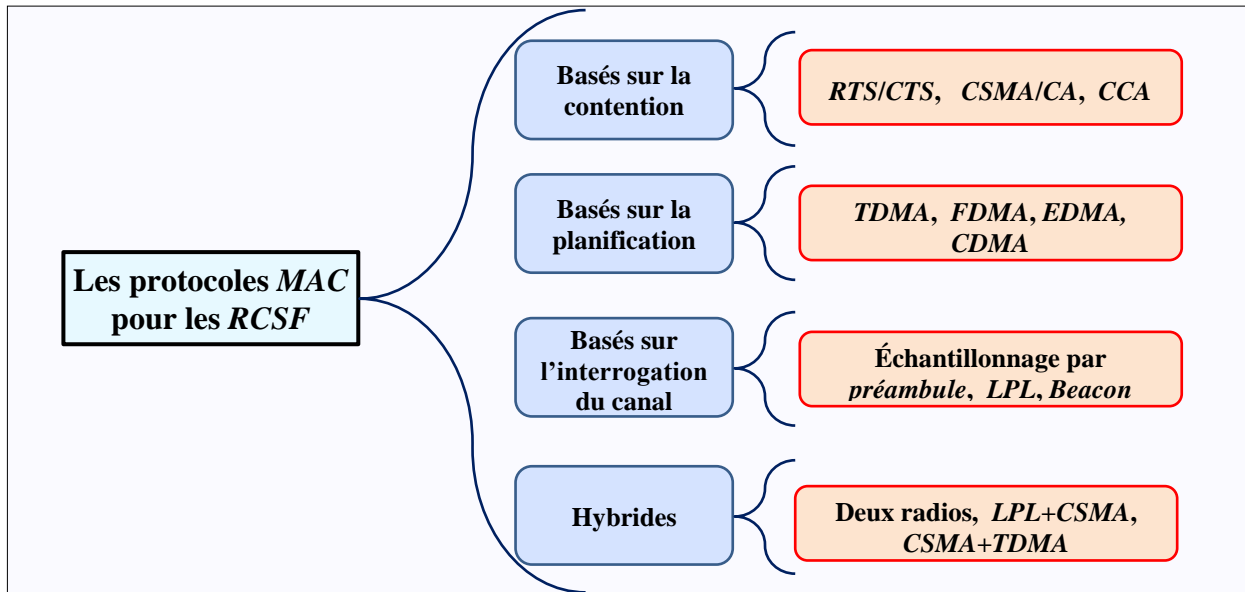


Figure 2.2: La classification des protocoles MAC selon le type d'accès au canal

2.3.2.1 Les protocoles MAC basés sur la contention

Dans les différents types de réseaux sans fil, le support de diffusion sans fil nécessite un procédé d'accès au canal efficaces pour arbitrer l'accès au médium, de sorte que la communication sans collision entre les nœuds puisse être offerte. D'une façon générale, les deux principales catégories d'accès au canal sont ; avec contention et sans contention [59] [60]. Dans les réseaux basés sur la contention, les périphériques se concurrencent pour accéder au canal. Les réseaux sans contention programment le canal, dans le temps ou la fréquence, où les appareils ne peuvent accéder qu'aux slots de canaux qui leur sont alloués, et communiquent ainsi avec le nœud central sans collision. Dans cette partie nous nous intéressons aux protocoles basés sur la contention. Dans ces derniers, les nœuds accèdent au médium pendant la même période de temps à l'aide d'un algorithme de la famille *CSMA/CA*, chacune de ses variantes tentant d'éviter les collisions. Ces protocoles sont caractérisés par l'extensibilité et le passage à l'échelle. Afin de bien comprendre l'accès au canal avec contention, le parfait exemple est l'algorithme *CSMA/CA* de la norme *IEEE 802.11* en mode *DCF (Distributed Coordination Function)* brièvement exposé dans la suite.

- ***CSMA/CA de la norme IEEE 802.11.*** Le schéma *CSMA/CA (Carrier Sence Multiple Access with Collision Avoidance)* fait partie de la même famille que *CSMA/CD (Carrier Sence Multiple Access with Collision Detection)* [61] car il oblige un expéditeur de s'assurer que le canal est inoccupé avant de commencer l'émission. Dans *CSMA/CA*

les collisions ne peuvent pas être détectées comme dans *CSMA/CD*, un nœud tente d'éviter les collisions mais ne pourra le faire à 100 % et cela est dû à l'effet aveuglant du médium sans fil (*Near Far Effect*) qui empêche un nœud de recueillir des données quand il est en train d'émettre. Dans la norme *IEEE 802.11* le fonctionnement est basé sur deux modes [62].

1. **Le premier est le mode infrastructure**, où les dispositifs communiquent entre eux à travers un point d'accès (appelé un nœud central). Dans ce mode deux techniques d'accès au canal peuvent être employé : la technique *DCF* avec le *CSMA/CA* et la technique *PCF (Point Coordination Function)*.

2. **Le second est le mode sans infrastructure (le mode ad hoc)**, où aucun point d'accès n'est utilisé par les dispositifs lors de leur communication entre eux. La seule technique qui peut être utilisée dans ce mode est le *DCF* avec le *CSMA/CA* dont l'accès au canal est structuré suivant une structure temporelle nommée super-trame et délimitée par des *beacons*.

▪ **La vérification de la disponibilité du canal par l'utilisation du mécanisme « Clear Channel Assessment ».** Avant qu'un expéditeur ne décide d'émettre, il doit contrôler au préalable que le canal de transmission est libre dans son environnement. Ce contrôle de la disponibilité du canal se réalise au moyen d'un mécanisme fourni par la couche physique appelé détection physique de porteuse ou *CCA (Clear Channel Assessment)*. Le fonctionnement de ce mécanisme est basé sur l'utilisation de deux seuils (voir la figure .2.3).

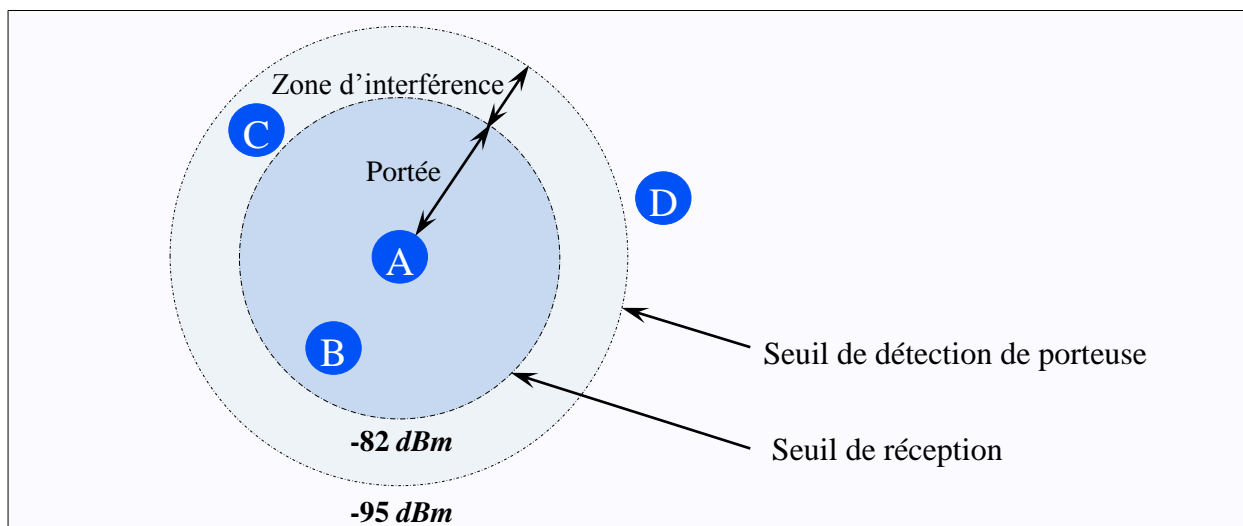


Figure 2.3: Les différents seuils utilisés par le mécanisme CCA

1. **Le premier est appelé le seuil de détection de porteuse (ou seuil d'interférence)**, est utilisé pour la vérification de la disponibilité du canal qui ne dépend pas du débit de transmission. Si la puissance détectée est supérieure à ce seuil, alors le canal est considéré occupé.

2. **Le second est appelé le seuil de réception**, qui est utilisé pour définir la portée d'une station et qui dépend essentiellement du débit de transmission. Si le niveau

de puissance est inférieur à ce seuil, alors la trame n'est plus décodable par le récepteur. Dans la norme *IEEE 802.11*, avec un débit de 11 *Mbits/s* le seuil de réception est égal à -82 dBm et le seuil d'interférence est égal à -95 dBm .

La figure 2.3 illustre un exemple de quatre nœuds *A*, *B*, *C* et *D*, où le nœud *A* peut transmettre très efficacement les signaux au nœud *B*, qui ne peuvent pas être déchiffrés par le nœud *C* et qui le gênent. Par contre le nœud *D* n'aperçoit aucun signal particulier montrant que le nœud *A* est en train d'émettre.

▪ **Accès au médium.** Un expéditeur détecte d'abord le support pour une période *IFS* du *DCF*, appelée *DIFS* (*DCF InterFrame Spacing*). S'il s'avère que le support est inactif, l'expéditeur accède au canal, sinon si au contraire le canal est occupé, il retente une émission après une période aléatoire de *back off*. Cet algorithme calcule un temps aléatoire de retransmission selon le nombre de tentatives de transmission exécutées. Si le canal reste inactif pendant cette période, le nœud prend le relais et lance un temporisateur avec une valeur aléatoire dans la fenêtre de contention actuelle. Le temporisateur est décrémenté après chaque emplacement. Le premier nœud dont l'horloge atteint zéro commence la transmission. D'autres nœuds détectant cette transmission, gèlent leurs horloges d'arrêt, qui doivent être redémarrées à la fin de la transmission en cours pendant une durée *DIFS* au cours de la période de contention suivante. Après une durée *SIFS* (*Short InterFrame Spacing*), le nœud récepteur envoie une trame d'acquittement après la réception correcte de trame envoyée par un expéditeur.

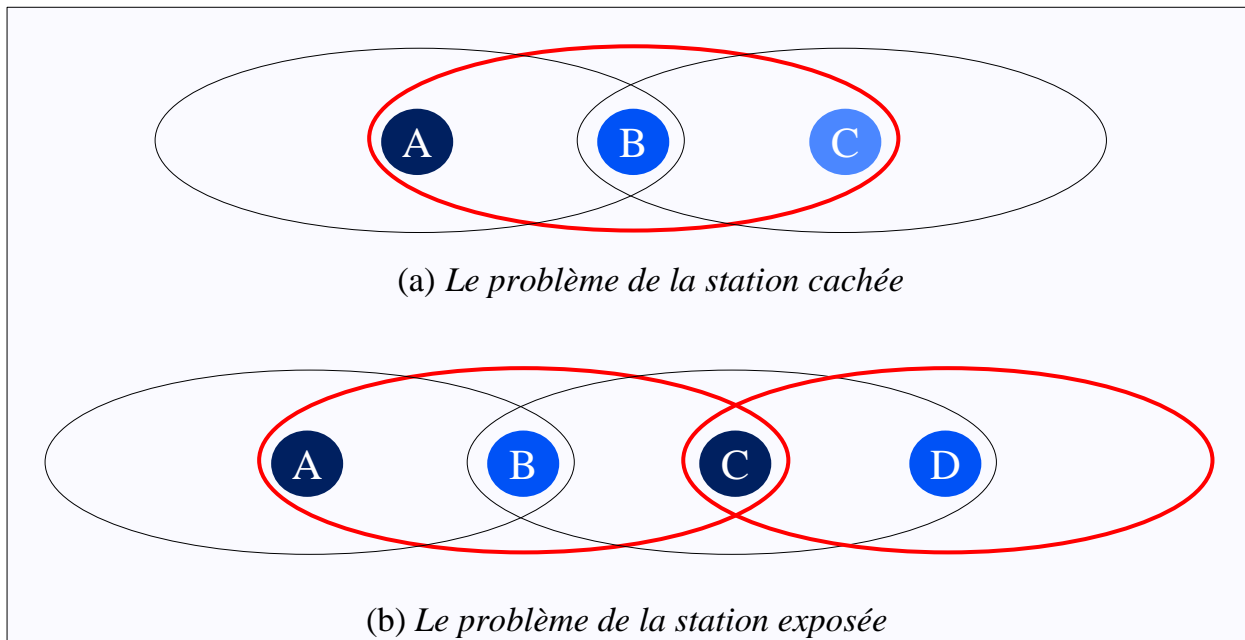


Figure 2.4: Illustration du problème de la station cachée/exposée [59]

▪ **Le problème de la station cachée et de la station exposée.** Comme le montre la figure 2.4(a), le problème de la station cachée se produit lorsque *C*, qui se trouve à la portée de *B*, veut lui transmettre des informations alors que *A* est déjà en communication avec *B*. Or, *C*, étant la station cachée, est hors de portée de *A* donc elle ne peut pas

s'apercevoir de la transmission de celle-ci. Ainsi, *C* trouve le canal libre et donc elle émet. Une collision se produit alors au niveau de *B*. Pour le problème de la station exposée, il s'agit du cas contraire. Dans la figure 2.4(b), *B* est en train d'émettre vers *A* et *C* veut émettre vers *D*. La dernière émission ne sera pas effectuée du fait que *C* se trouvant à la portée de *B* croit que le canal est occupé. Or, l'émission de *B* vers *A* ne causerait pas de problème car *D* n'est à la portée ni de *B* ni de *A*. Ainsi, la station *C*, exposée à l'émission de *B* vers *A*, est forcée d'attendre inutilement : le débit s'en ressent donc. C'est ainsi que se présentent les problèmes du CSMA/CA original auquel il a fallu ajouter un mécanisme adéquat permettant d'un côté de diminuer les collisions et d'un autre d'augmenter le débit et optimiser l'utilisation du canal.

Pour remédier au problème de la station cachée, la norme 802.11 utilise le mécanisme du RTS/CTS (*Request To Send/Clear To Send*). Dans le cas de la figure 2.4(a), quand *C* trouve le médium libre, avant d'envoyer les trames de données, il envoie une trame RTS à *B* et attend la réception de CTS provenant de *B*. De cette façon si *B* ne répond pas par un CTS, *C* n'envoie pas ses trames de données. L'écoute des messages RTS/CTS qui servent à réserver localement le canal en indiquant la longueur de la trame à venir permet de réduire les risques de collision.

▪ **Le protocole S-MAC (*Sensor Medium Acces Control*):** le protocole *Sensor-MAC* (*S-MAC*) [63] a été développé à l'université de la Californie Los Angeles, présente l'une des premières tentatives pour réduire la consommation d'énergie provoquée par l'écoute de la porteuse à vide (*idle listening*), les collisions, écoute abusive (*l'overhearing*), et l'utilisation des paquets de contrôle dans les RCSF en basculant les nœuds en mode veille autant que possible (tant que rien ne s'est produit). La durée de la période d'activation a normalement une taille fixe en fonction de certains paramètres de la couche *Physique* et *MAC*, tandis que la durée des périodes de veille dépend du paramètre de duty-cycle prédéfini basé sur l'application. *S-MAC* tente de coordonner les nœuds voisins en leur permettant de partager des périodes d'activation communes selon un calendrier. Cela nécessite la formation et le maintien de la synchronisation entre les nœuds.

La période d'activation est divisée en deux intervalles. La période *SYNC* et la période de données, comme indiqué sur la figure 2.5. Pendant la période *SYNC* et afin de réduire la latence des paquets de contrôle, un nœud essaie de recevoir un paquet *SYNC* de ses voisins, qui comprend *l'ID* de l'expéditeur et le temps restant jusqu'à ce que l'expéditeur passe en mode veille. Lors de l'initialisation, si le nœud n'entend pas le calendrier de ses voisins et avant qu'il bascule vers le mode veille, il choisit immédiatement son propre calendrier et le diffuse. Ce nœud s'appelle le synchroniseur et d'autres nœuds commencent à se synchroniser avec lui. Si le nœud trouve un calendrier, il stocke également ce dernier dans sa table de planification et il devient ainsi un nœud suiveur. La période de données est utilisée pour échanger des messages relatifs aux données, qui peuvent inclure des messages *RTS*, *CTS*, *DATA* ou *ACK*. Afin de minimiser l'effet de terminal caché, *S-MAC* utilise le schéma de négociation *RTS/CTS* pour les paquets de mono diffusion. Les paquets de diffusion sont envoyés sans utiliser les

messages de contrôles *RTS/CTS*. Pour minimiser les retransmissions et les collisions coûteuses et, *S-MAC* fournit un mécanisme de transmission de messages, dans lequel les messages longs sont fragmentés en courtes trames et envoyés en rafale. Le *RTS/CTS* est requis uniquement avant la transmission de la première courte trame.

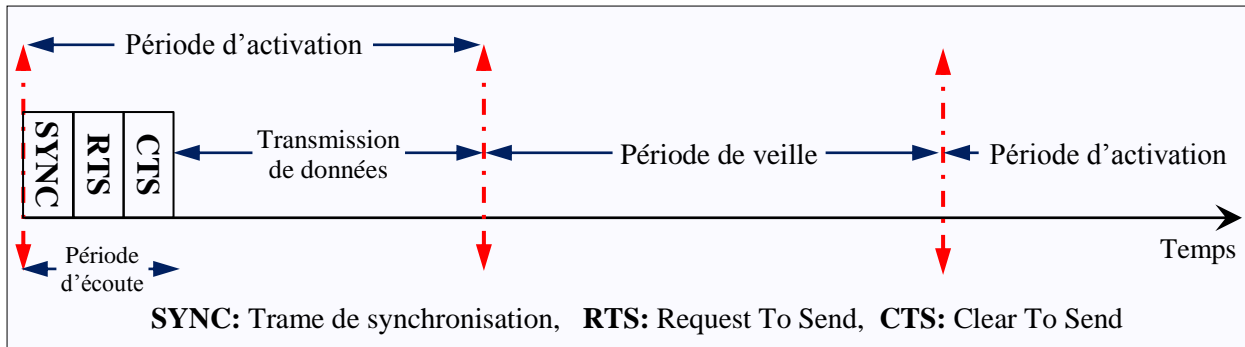


Figure 2.5: Les deux périodes « d'activation et de veille » périodique du *S-MAC* [63]

Avec *S-MAC*, un nœud peut recevoir plusieurs calendriers de ses voisins. Dans ce cas, il s'adapte à tous les calendriers et agit comme un nœud de bordure (ou nœud frontalier) pour deux ou plusieurs clusters virtuels. Ainsi, un réseau *S-MAC* est susceptible d'avoir plusieurs clusters virtuels. La figure 2.6 montre un exemple de formation de cluster virtuel avec *S-MAC*. Les nœuds frontaliers doivent se réveiller aux intervalles d'activation de tous ces horaires pour la formation réussie de grappes virtuelles. Comme un nœud tente de suivre les programmes existants pendant la période d'initialisation, *S-MAC* s'attend à ce que le nœud ne s'adapte que rarement à plusieurs programmes (des horaires multiples).

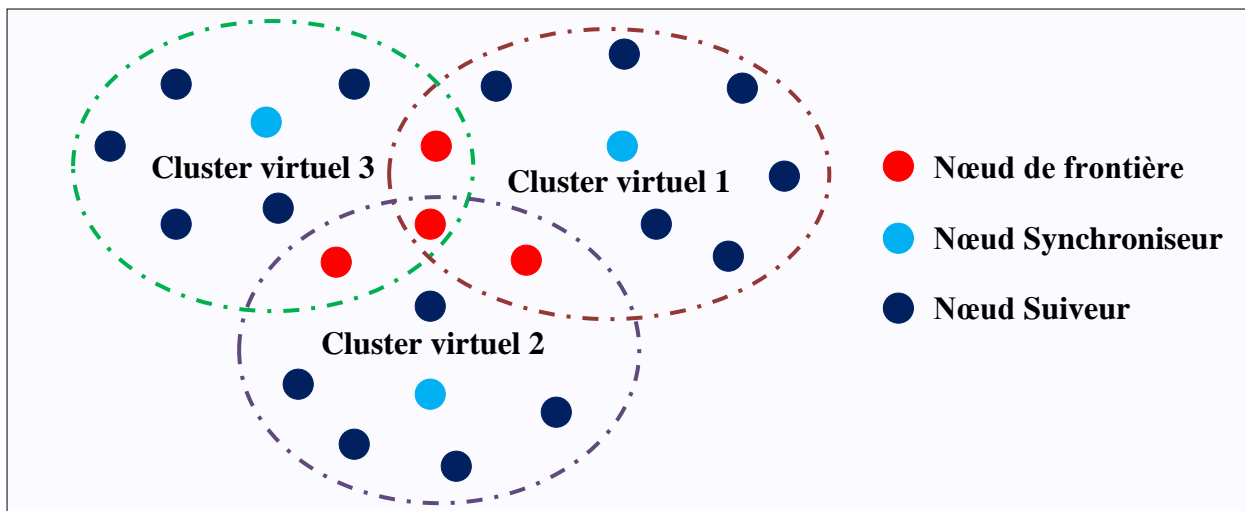


Figure 2.6: La formation des clusters virtuels avec *S-MAC* [45]

Bien que le *S-MAC* réduise l'écoute à vide (*idle listening*) et l'écoute abusive (*Overhearing*) dans le duty-cycle des nœuds, il présente plusieurs autres inconvénients. *S-MAC* est rigide et optimisé pour un ensemble prédéfini de charges de travail, car il n'est pas possible d'adapter la durée des périodes d'écoute et de veille à l'évolution des conditions de trafic. Comme mentionné précédemment, la formation, la maintenance et la

conformité de la synchronisation ont de graves conséquences sur les RCSF. Les périodes de veille plus longues et fixes de *S-MAC* ont un impact considérable sur la latence du système. Cela pourrait s'aggraver si les nœuds intermédiaires sur une route ne partagent pas une planification commune. Pour réduire la latence, un mécanisme d'écoute adaptatif est utilisé avec *S-MAC*. Un nœud x , qui aperçoit la transmission en cours de ses voisins, reçoit le paquet *RTS/CTS*, extrait le temps restant de la transmission du champ de durée du paquet *RTS/CTS* et se réveille juste à la fin de la transmission. Le nœud reçoit et traite le paquet si le paquet lui est destiné, sinon il repasse en mode veille. Cependant, une fois que le nœud x reçoit le paquet, il doit attendre la prochaine période d'écoute pour trouver le prochain saut et transmettre le paquet plus avant, ce qui limite les avantages de l'écoute adaptative. La consommation d'énergie peut être augmentée avec l'écoute adaptative, car tous les nœuds en écoute ont besoin de se réveiller à la fin de la transmission en cours.

Les nœuds frontaliers dépensent plus d'énergie que les nœuds non-frontaliers au fur et à mesure qu'ils se réveillent à chaque échancier afin de maintenir la connectivité réseau entre les clusters. *S-MAC* ne permet pas de contrôler la taille des grappes virtuelles et le nombre de nœuds de bordure. Avec la fragmentation dans le *S-MAC*, la surcharge et la retransmission peuvent être réduites, mais cela induit de l'injustice, puisque le nœud réserve le canal pendant toute une durée de rafale. Un nœud voisin transportant des données liées à un retard devra attendre plus longtemps pour accéder au canal.

▪ **Le protocole *T-MAC (Time-out MAC)***: le protocole *T-MAC (Time-out MAC)* [64] améliore l'efficacité énergétique du protocole *S-MAC*, en particulier dans des conditions de trafic variables, en raccourcissant de manière adaptative les périodes d'écoute des nœuds de capteurs. Contrairement au *S-MAC*, où les périodes d'écoute sont toujours rigides, la période d'écoute d'un nœud *T-MAC* se termine lorsqu'aucun événement d'activation ne s'est produit pour une période de seuil TA . Cet aspect dynamique libère l'application de la sélection d'une valeur de duty-cycle approprié. La comparaison entre le *S-MAC* et le *T-MAC* présentée à la figure 2.7 confirme cette amélioration. Cependant, cette amélioration dépend fortement de la valeur du TA . Ce dernier doit être suffisamment long pour qu'un nœud puisse détecter la porteuse et entendre un *CTS* potentiel d'un voisin. Bien que le *T-MAC* fonctionne mieux sous des charges variables, la synchronisation des périodes d'écoute au sein de grappes virtuelles peut être partiellement interrompue. Cela pourrait entraîner un problème du sommeil précoce pour les nœuds avec *T-MAC*. Le sommeil précoce se produit lorsqu'un nœud, en particulier le troisième saut, passe en mode veille alors qu'un voisin a encore des messages pour lui. Cependant, avec *T-MAC*, les trames *FRTS (Future Request To Send)* peuvent être envoyées aux nœuds du troisième saut pour prolonger la durée de TA ou pour les laisser se réveiller au moment approprié. Le *T-MAC* économise plus d'énergie que le *S-MAC* et minimise les collisions et la redondance, car les nœuds inactifs reviennent relativement plus tôt au mode veille. Cependant, cela se traduit par un débit réduit et une latence réseau supérieure. *T-MAC* souffre également de problèmes de synchronisation et de mise à l'échelle.

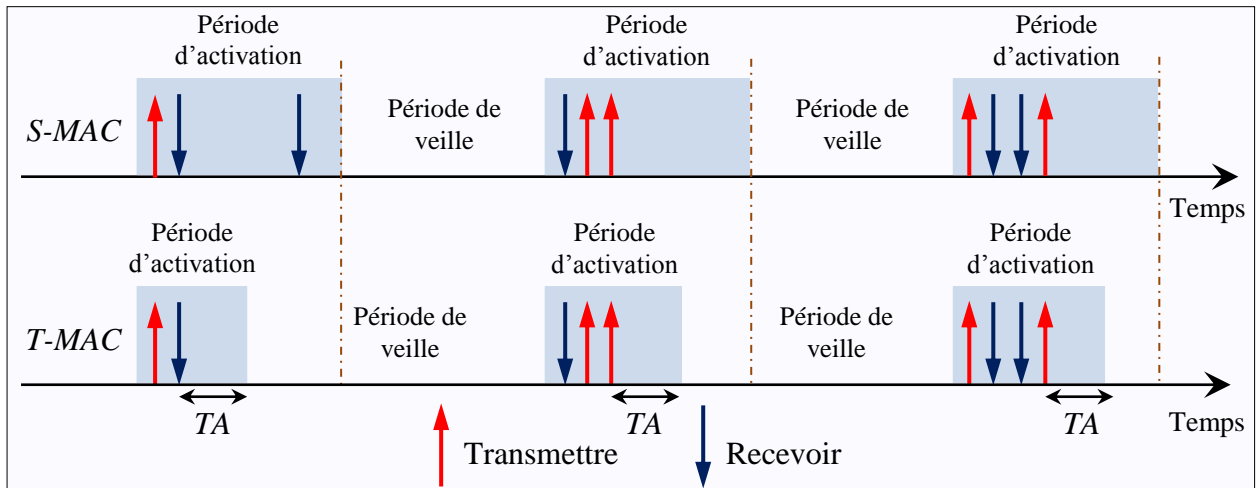


Figure 2.7: Les deux périodes «d'activation et de veille» périodique du S-MAC et T-MAC [125]

▪ **Le protocole DMAC (Data gathering MAC) :** le protocole *Data gathering MAC* (DMAC) [65] résout le problème de latence pour le modèle de communication de conversion convergée. DMAC a été conçu à l'origine pour améliorer S-MAC. L'idée de base consiste à échelonner les temps actifs en fonction du niveau de l'arbre de recouvrement, de sorte que les données puissent passer rapidement des feuilles à la racine, voir la figure 2.8. Chaque nœud écoute d'abord ses enfants, puis propage tous les messages à son parent. DMAC utilise un simple CSMA avec des accusés de réception. Les nœuds perdant la contention n'ont pas besoin d'attendre le prochain flux vers le haut, mais peuvent réessayer dans un intervalle trop long programmé après une paire *Recv/Send* (*Recevoir/Envoyer*) occupée. Pour tenir compte des interférences avec le trafic plus haut dans l'arborescence, ces paires excessives sont planifiées avec un intervalle de 3 slots. Les slots de débordement augmentent essentiellement la capacité à la demande, ce qui permet à DMAC de s'adapter automatiquement à la charge de trafic, tout comme la prolongation de la période active par T-MAC.

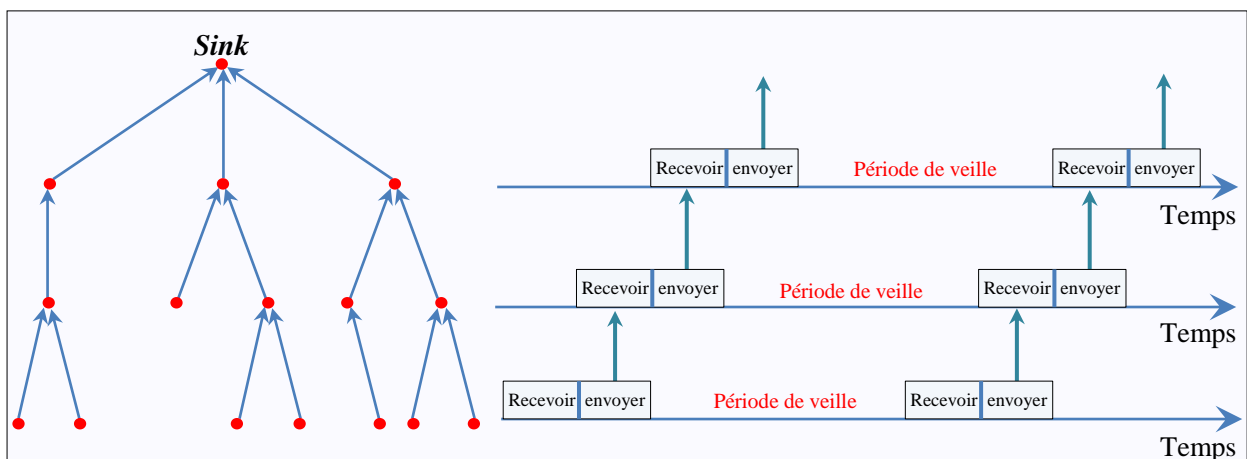


Figure 2.8: Arbre de conversion convergée avec correspondance, emplacements DMAC décalés [65]

L'inconvénient de DMAC est qu'il lui est impossible de prendre en charge des modèles de communication autres que la conversion convergée. En particulier, les

commérages locaux basés sur la diffusion ne fonctionnent pas parce que les voisins (enfants, pairs et parents) écoutent à des moments différents. Cela pourrait bien être la raison pour laquelle *DMAC* n'a jamais réussi l'étape de simulation. Néanmoins, l'étalement des périodes actives est une approche convaincante pour réduire la latence sans augmenter la consommation d'énergie, ce qui justifie des recherches plus poussées.

2.3.2.2 Les protocoles MAC basés sur la planification

Au cours de la phase d'initialisation, les schémas basés sur la planification attribuent des liaisons sans collision aux nœuds voisins. Cependant, les liaisons peuvent être attribuées sous plusieurs formes selon différents schémas. En raison de la complexité de quelques schémas tels que : *FDMA* et *CDMA*, les *RCSF* préfèrent donc les schémas *TDMA* en tant que méthodes de planification afin de réduire la complexité engendrée [66]. Dans les schémas *TDMA*, l'heure système est divisée en slots. Ces slots sont ensuite attribués à tous les nœuds voisins pour lui garantir l'accès au canal. Cependant, le programme contrôle l'autorisation du participant sur les ressources avec un temps réglementaire. Le programme est généralement réglementé par une autorité centrale; de plus, il peut être fixe ou calculé à la demande (ou hybride). D'autre part, un nœud n'a pas besoin de conflit avec ses voisins, car il ne peut accéder qu'à son slot de temps attribué.

Les principaux avantages de ces schémas sont : la minimisation de la collision, la réduction de *l'overhearing* et l'évitement implicite de l'écoute à vide (*idle listening*). De ce fait, les systèmes basés sur la planification fournissent également un délai de bout en bout limité et prévisible. Le délai moyen de mise en file d'attente est normalement élevé, car le nœud doit attendre son slot de temps attribué avant d'accéder au canal. Cependant, ces schémas ont d'autres problèmes tels que : *l'overhead* et le trafic supplémentaire, manque d'adaptabilité, une scalabilité réduite et un faible débit. De plus, dans ces schémas, l'allocation de planification *TDMA* sans conflit est une tâche vraiment difficile. Cependant, les protocoles *MAC* basés sur *TDMA* ont attiré l'attention des chercheurs du *RCSF*. Dans ce qui suit certains protocoles sont présentés.

▪ **Le protocole *TRAMA* (*TRaffic-Adaptive MAC*) [67]** : est basé sur *TDMA*. Il crée des planifications pour les nœuds synchronisés dans le temps de manière distributive en fonction des informations de trafic. Le temps du système est divisé en cycles, chacun contenant les périodes d'accès aléatoire et les périodes d'accès planifié. La période d'accès aléatoire consiste en un ensemble de tranches de signalisation, tandis que la période d'accès planifiée contient un ensemble de slots de transmission de données, comme illustré à la figure 2.9. Pour ce faire, *TRAMA* utilise trois mécanismes :

1. **le protocole de voisinage (*NP* (*Neighbor Protocol*))** pour échanger la liste de voisins à un saut entre les nœuds, et ainsi d'établir une connaissance de topologie des voisins à deux sauts, il fonctionne pendant la période d'accès aléatoire.
2. **le protocole d'échange de calendrier (*SEP* (*Schedule Exchange Protocol*))** où chaque nœud échange son calendrier de transmission courant et réceptionne également les calendriers de ses voisins. Dans le but d'avoir un slot de temps, le

nœud calcule sa propre priorité et la priorité de tous ses voisins à deux sauts pour chaque slot de temps.

3. l'algorithme d'élection adaptatif (AEA (*Adaptative Election Algorithm*)) est utilisé pour sélectionner les émetteurs et les récepteurs à partir des slots de temps actuels sur la base des informations de voisinage et de planification collectées par le protocole *NP* et le protocole *SEP* respectivement. Cet algorithme fonctionne pendant l'accès planifié.

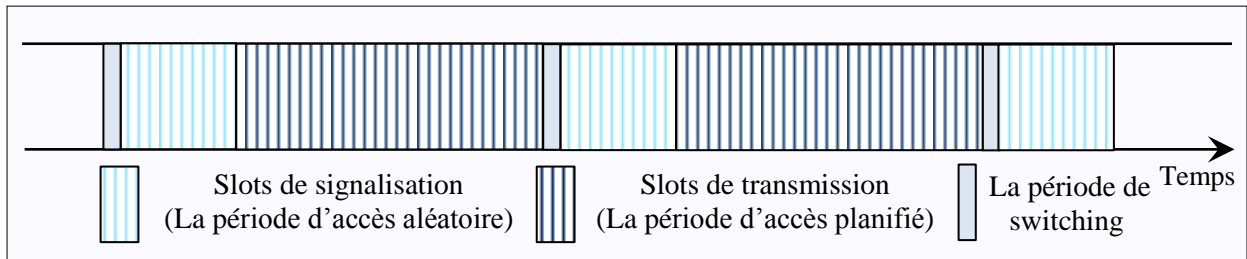


Figure 2.9: Les deux périodes « d'accès aléatoires et d'accès planifiés » dans *TRAMA* [67]

Un nœud utilisant *TRAMA* peut avoir l'un des trois états suivants: transmission, réception et veille. L'état du nœud est déterminé en fonction des informations de voisinage à deux sauts et des calendriers annoncés par ses voisins à un saut. Le nœud passe à l'état de transmission s'il a des données à envoyer avec la priorité la plus élevée parmi ses concurrents. Sinon, le nœud consulte l'horaire envoyé par l'émetteur actuel. Si le trafic de l'émetteur est destiné à ce nœud dans le slot actuel, il reste en mode réception, sinon il passe en mode veille. Ainsi, *TRAMA* se présente comme le premier protocole à introduire l'aspect économie d'énergie dans son mécanisme de découpage temporel en mettant en mode veille les nœuds non concernés par une transmission ou une réception durant un slot de temps donné. Il améliore aussi le débit utile en évitant les collisions des trames de données.

▪ **Le protocole *FLAMA* (*Flow-Aware MAC*) [68]:** est dérivé du protocole *TRAMA*, et optimisé pour les applications de surveillance périodique. L'idée principale de *FLAMA* est d'éviter l'utilisation des messages de contrôles (*Overhead*) liés à l'échange d'informations sur le trafic. *FLAMA* collecte des informations sur le flux des données des nœuds voisins à un seul saut et établit une connaissance sur la topologie des nœuds à deux sauts. Il divise le temps suivant en deux périodes d'activité : période d'accès planifié et période d'accès aléatoire.

D'un autre côté, *FLAMA* ne diffuse pas de calendriers de trafic pendant les périodes d'accès planifié, mais échange des informations par rapport aux flux de données liés à l'application pendant les périodes d'accès aléatoire. En outre, *FLAMA* établit une synchronisation globale du réseau basé sur l'estampillage des trames et une topologie arborescente pour le relais de données. Pour déterminer les émetteurs et les récepteurs concernés par chaque slot de temps lors de la période d'accès planifié, *FLAMA* applique un algorithme plus simple que celui utilisé dans *TRAMA*. Comme *TRAMA*, *FLAMA*

assure des transmissions sans collision en ne permettant qu'à un seul nœud d'émettre dans un voisinage à deux sauts.

FLAMA surpasse *TRAMA* en terme d'économie d'énergie et de perte de trames. De plus, *FLAMA* améliore le délai de bout en bout par rapport à *TRAMA* grâce à l'exploitation des chemins définis à partir d'un arbre pour le cheminement multi-sauts. En revanche, *FLAMA* surcharge le réseau par des échanges pour connaître le flux de l'application et établir une synchronisation globale du réseau de proche en proche.

2.3.2.3 Les protocoles MAC basés sur l'interrogation du canal

Le système d'interrogation de canal est appelé échantillonnage de préambule (*LPL* (*Low Power Listening*)). Le nœud envoie un paquet de préambule sur le canal pour garantir que le nœud de destination détecte l'activité radio et se réveille avant de recevoir les données de l'émetteur. Au réveil, si une activité radio est détectée par le récepteur, le récepteur allume sa radio pour recevoir des paquets de données. Sinon, le nœud (récepteur) bascule en mode veille jusqu'au prochain intervalle d'interrogation. Cette vérification doit être effectuée aussi longtemps que la durée de l'intervalle de vérification jusqu'à l'envoi du préambule [69]. Contrairement aux deux familles précédentes de protocoles *MAC*, les protocoles fonctionnant sur le principe de l'échantillonnage de préambule n'ont pas besoin que les nœuds soient synchronisés pour communiquer. Cette catégorie de protocoles [70] [71] n'a pas besoin d'utiliser de périodes communes d'activités et de veille. À la place, chaque nœud est laissé libre de choisir son cadencement indépendamment des autres nœuds du réseau.

▪ **Le protocole *BMAC* (*Berkeley MAC*) [18]:** l'un des premiers protocoles *MAC* pour les *RCSF* basés sur la technique d'échantillonnage de préambule est *BMAC* (*Berkeley MAC*). Pour s'assurer que le récepteur est prêt à réceptionner des données, un nœud émetteur va transmettre un long préambule où leur durée de transmission est plus longue que la période de veille. L'émetteur est ainsi assuré que tous les nœuds dans son voisinage, notamment le récepteur, sont réveillés et prêts à recevoir les données. Un mot de synchronisation est envoyé afin d'indiquer la fin du préambule et la proximité de l'envoi du paquet de données. Les nœuds écoutent périodiquement le canal pendant un laps de temps très court noté *CCA* afin de vérifier si une transmission radio est en cours. Le temps d'attente d'un nœud entre deux *CCA* correspond à l'intervalle de vérification noté *CI* (*Check Interval*). Avant d'émettre ses données, un émetteur doit être sûr que le destinataire est réveillé. Pour cela, la durée du préambule doit nécessairement être aussi longue que l'intervalle de vérification *CI* du récepteur. Ainsi, l'émetteur est certain que le récepteur sera réveillé pour réceptionner les données. Suivant les fonctionnalités fournies par le protocole *MAC*, un acquittement (*Ack* (*Acknowledgement*)) peut être transmis par le récepteur de la trame de données pour prévenir l'émetteur de la bonne réception des données. L'inconvénient de *BMAC* est la forte écoute des nœuds voisins au nœud émetteur qui n'étant pas le récepteur jusqu'à la réception de la trame de données qui

génère une consommation d'énergie inutile. La figure 2.10 montre le fonctionnement du protocole *BMAC*.

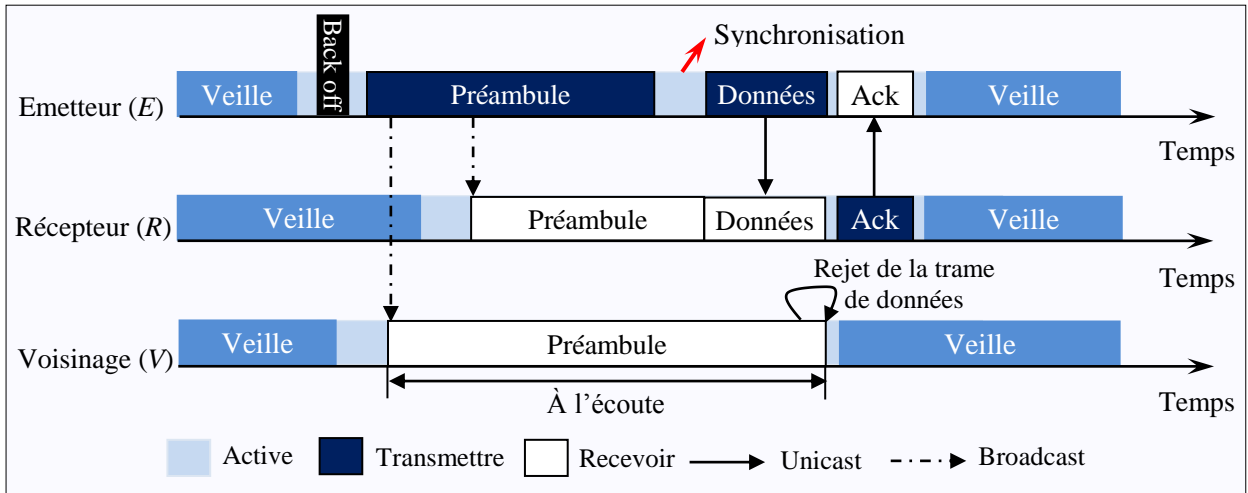


Figure 2.10: La transmission des données avec le protocole *BMAC* [125]

▪ *Le protocole XMAC (A short preamble MAC protocol for duty-cycled wireless sensor networks)* [19]: *XMAC* améliore le *BMAC*, en divisant la longue trame de préambule utilisée par le *BMAC* en courtes trames de préambules contenant l'adresse du récepteur prévu de la trame de données; le nœud émetteur transmet ces courtes trames de préambules, avec une petite période d'écoute après chaque transmission, comme illustré à la figure 2.11. Lorsque le récepteur se réveille et entend une courte trame de préambule indiquant qu'il s'agit du récepteur prévu, il envoie un paquet *ACK* précoce, pour demander au nœud émetteur de commencer immédiatement à transmettre la trame de données. Cela permet d'économiser du temps d'attente à la fois sur les nœuds émetteur et récepteur et permet aux autres nœuds voisins éveillés de basculer au mode veille. La simplicité de ces protocoles les a rendues populaires et différentes propositions ont été soumises pour réduire leur consommation énergétique. Ainsi, que l'émission de ce préambule est cependant coûteuse ce qui rend ces protocoles plus adaptés à des trafics de données irréguliers.

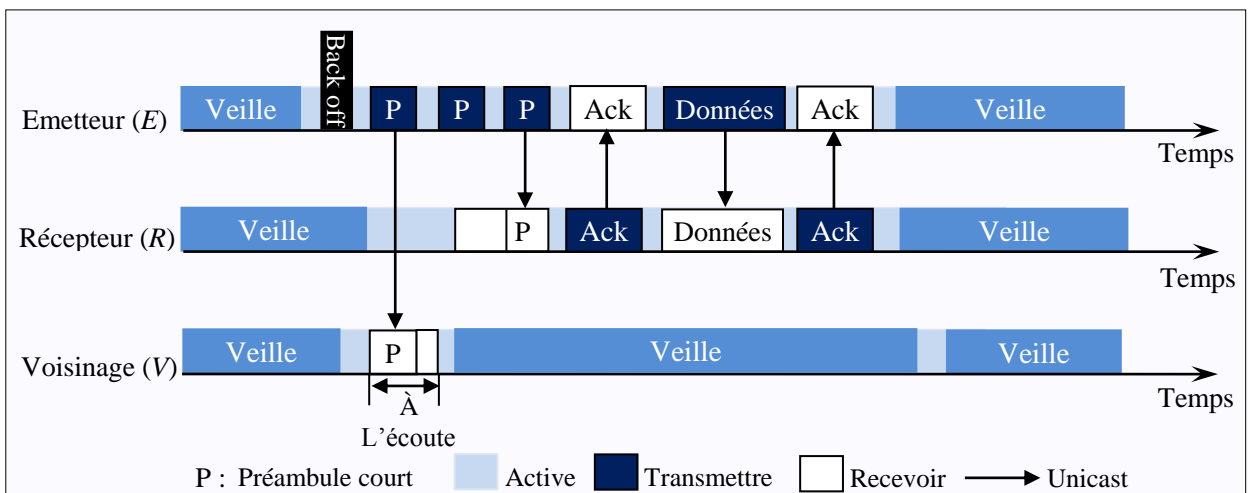


Figure 2.11: La transmission des données avec le protocole *XMAC* [20]

▪ **Le protocole *RI-MAC (Receiver-Initiated MAC)* [20]:** malgré que le protocole *RI-MAC* [90] ne soit pas vraiment un protocole d'échantillonnage de préambule, la raison de la classer dans cette catégorie est que les auteurs ne l'ont comparée qu'avec les deux protocoles d'échantillonnage de préambule; *BMAC* et *XMAC*. Le *RIMAC* diffère des autres protocoles en ce qui concerne la communication entre l'émetteur et le récepteur. Avec le *RI-MAC* la communication est initiée par le nœud récepteur où chaque nœud se réveille périodiquement en fonction de son propre programme pour vérifier s'il existe des trames de données entrantes destinées à ce nœud. Après avoir allumé sa radio, un nœud diffuse immédiatement une balise si le support est inactif, annonçant qu'il est réveillé et prêt à recevoir une trame de données. Un nœud émetteur *E* avec les données en attente à envoyer reste actif silencieusement dans l'attente de la balise du récepteur prévu *R*. A la réception de la balise à partir de *R*, le nœud *E* lance immédiatement sa transmission de données qui sera reconnue par *R* avec une autre balise *ACK*. Notez que cette balise *ACK* a deux rôles: pour le début, elle accuse la réception de la trame de données, puis elle invite une nouvelle transmission de trame de données au même récepteur *R*. S'il n'y a pas de données entrantes après la diffusion d'une balise, le nœud se met en veille, comme illustré à la figure 2.12. Comme aucun préambule n'est transmis sur le canal, seule la transmission de balise et de paquet de données, *RI-MAC* peut limiter l'utilisation du canal par rapport aux protocoles d'échantillonnage du préambule. Les balises transmises peuvent également aider à la coordination des voisins et, en conséquence, le récepteur fait varier de manière adaptative l'utilisation de son canal en fonction des charges de trafic. Cela permet au *RI-MAC* d'atteindre un débit équitable, un taux de livraison des paquets et une efficacité énergétique dans des charges de trafic variables.

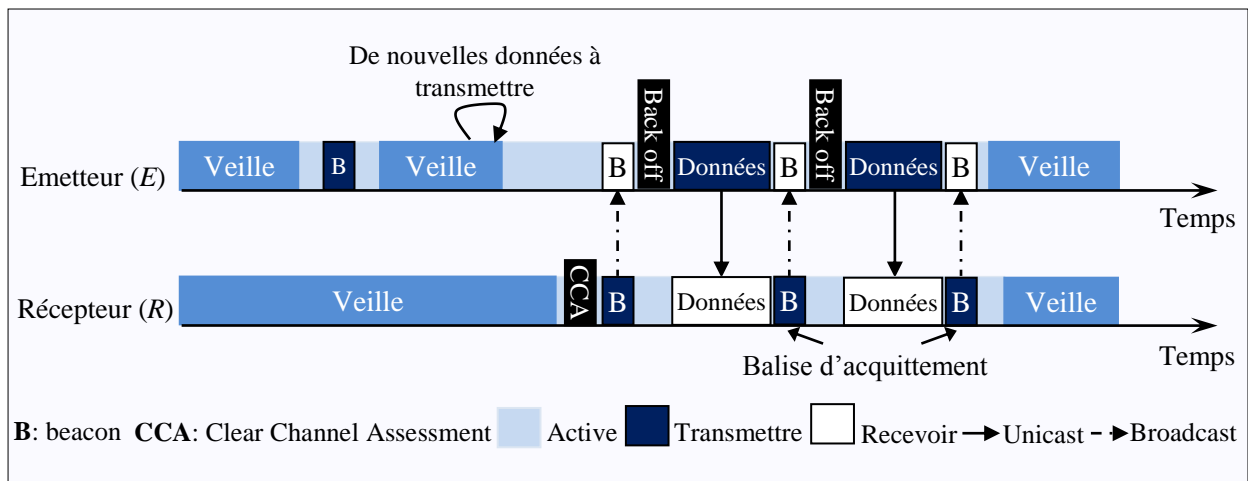


Figure 2.12: La transmission des données avec le protocole *RI-MAC* [20]

Bien que l'idée de transférer l'initiation de la communication de l'expéditeur au destinataire soit intéressante, ces systèmes sont en fait similaires à la méthode classique, avec seulement la différence de transfert de la charge de communication d'une partie à une autre. Par conséquent, ces protocoles portent tous les problèmes discutés de l'écoute à vide (*idle listening*) et l'écoute abusive (*Overhearing*) et de la collision d'une manière ou d'une autre.

2.3.2.4 Les protocoles MAC Hybrides

Afin de parvenir à une amélioration conjointe, les protocoles MAC hybrides combinent les forces de deux ou plusieurs schémas MAC différents. En général, les protocoles MAC hybrides combinent un système synchronisé avec un système asynchrone. Bien que les protocoles hybrides cumulent les avantages de plusieurs systèmes, ils peuvent également entraîner des problèmes de complexité et de mise à l'échelle dans le maintien de deux modes de fonctionnements différents ou plus. Cependant, le protocole ZMAC (Zebra MAC) [72], le protocole SCP-MAC (The Scheduled Channel Polling MAC) [73] et le protocole Funneling-MAC [74] sont des exemples importants sur ce schéma.

- **Le protocole ZMAC (Zebra MAC):** est un protocole hybride qui bascule les périodes de TDMA et CSMA/CA en fonction du nombre d'entités en concurrence à un moment donné. ZMAC est un protocole adaptatif au trafic dans le sens où, en cas de faible contention, il bascule vers le CSMA pour obtenir une utilisation élevée du canal et des délais faibles, et en cas de contention élevée, il passe à TDMA afin d'atteindre une utilisation élevée du canal, l'équité et moins de collision.

Le protocole ZMAC commence par exécuter un algorithme d'attribution de slot distribué qui prend en compte le voisinage à deux sauts pour arriver à un planning sans conflit. Ensuite, les nœuds doivent se disputer l'accès pour envoyer un message. Par défaut, un nœud peut se battre pour n'importe quel emplacement, mais un propriétaire obtient la priorité en affrontant en premier, voir la figure 2.13. Lorsqu'un nœud constate qu'il perd trop de paquets, il diffuse un paquet de notification explicite qui demande aux nœuds de passer en mode de conflit élevé. Dans ce mode, les nœuds ne peuvent plus faire face aux emplacements appartenant à leurs voisins de second saut, ce qui les empêche de provoquer d'autres collisions en tant que terminaux cachés. Après 10 secondes, les nœuds de temporisation reviennent au fonctionnement normal. Le ZMAC construit essentiellement une version TDMA au-dessus du CSMA de base, en utilisant le BMAC (Low-power listening (écoute de faible puissance)) pour obtenir une efficacité énergétique. Un avantage supplémentaire de la superposition au CSMA est qu'elle rend le ZMAC très tolérant à la dérive de l'horloge, contrairement à de nombreux autres schémas TDMA.



Figure 2.13: Structure de slot avec préférence intégrée pour les propriétaires de slot [68]

ZMAC est plus performant que BMAC sous forte et moyenne charge, et moins performant sous faible charge, surtout en termes d'économie d'énergie. ZMAC est fait pour optimiser l'utilisation du canal en se comportant comme CSMA/CA sous faible

charge et comme *TDMA* sous forte charge. Le *CSMA/CA* utilisé par *ZMAC* n'est pas économe en énergie, car chaque nœud écoute le médium tant qu'il est détecté occupé. *ZMAC* n'emploie pas un mécanisme de report afin de ne pas déborder sur le slot suivant et ne garantit pas qu'un seul nœud soit propriétaire d'un slot. Cela est dû aux imprécisions de synchronisation et aux erreurs potentielles d'assignation de slots. Une réutilisation de slots à partir de deux sauts génère des collisions à cause des acquittements de niveau *MAC*.

- **Le protocole *SCP-MAC* (*The Scheduled Channel Polling MAC*):** est un autre protocole hybride qui utilise l'échantillonnage de canal avec *TDMA* et permet d'obtenir une longueur de préambule réduite. L'interrogation périodique du canal avec des heures d'échantillonnage synchronisées et des nœuds dormant sans trafic sur un canal utilisant ce protocole. *SCP-MAC* utilise deux phases de contentions distinctes pour obtenir une faible probabilité de collision. Les résultats présentés dans [73] suggèrent que *SCP-MAC* a une durée de vie comparativement élevée d'un facteur 2-2.5.

Le fonctionnement de base de *SCP-MAC* est illustré à la figure 2.14. Avant de communiquer, le nœud attend en silence en mode veille et effectue la détection de la porteuse dans la première fenêtre de contention *CW1* juste avant l'heure d'interrogation du récepteur. Lorsqu'il trouve le canal inactif, l'expéditeur envoie un bref signal de réveil pour activer le récepteur. Sinon, il se met en veille et effectue une interrogation régulière des canaux. Une fois que l'émetteur a réveillé un récepteur, il entre dans la deuxième fenêtre de contention *CW2* et envoie des données si le nœud détecte que le canal est toujours libre. Le nœud récepteur qui a reçu une tonalité de réveil étend sa période de réveil pour recevoir le paquet de données.

Grâce à l'utilisation de petits préambules, il peut encore réduire le délai et le duty-cycle par un facteur de 10. Ce protocole peut également activer ou désactiver la prise de contact *RTS/CTS*. Lorsque *SCP-MAC* désactive l'établissement de la liaison et examine l'adresse de destination à partir de l'en-tête du paquet, ce qui entraîne une écoute abusive (*Overhearing*). Ce protocole a également une fonctionnalité d'écoute adaptative qui aide le canal à gérer un trafic supplémentaire. L'utilisateur peut consulter toutes les planifications des autres nœuds et sélectionner l'allocation de planification de raccourci pour éviter tout retard basé sur la planification. À ce stade du protocole *SCP-MAC*, le principal défi consiste à gérer une latence et une consommation d'énergie plus élevées en raison de l'échantillonnage de canaux et des modes *TDMA*.

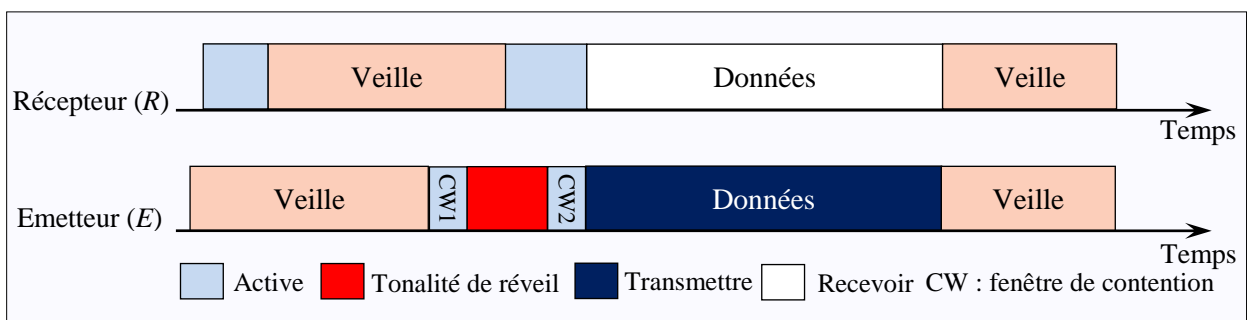


Figure 2.14: La transmission des données avec le protocole *SCP-MAC* [73]

▪ **Le protocole *Funneling-MAC*** : est principalement un protocole *CSMA/CA* avec un algorithme *TDMA* localisé, qui ne fonctionne qu'à quelques sauts du nœud récepteur dans la région d'intensité. Le fonctionnement de base de ce protocole est illustré à la figure 2.15. La planification de *TDMA* est gérée par le nœud récepteur et fonctionne localement dans la région d'intensité. La charge de calcul et de maintien de la profondeur de la région d'intensité tombe également sur le *sink*. Le nœud récepteur diffuse une beacon qui déclenche la planification de *TDMA*. Tous les nœuds exécutent *CSMA/CA* par défaut, à moins qu'ils ne reçoivent de beacon et soient ensuite considérés comme étant dans la région d'intensité. Cette dernière est définie par le nœud récepteur en contrôlant la puissance de transmission de beacon. Les nœuds qui ne reçoivent pas de beacon dans le délai spécifié qui se désignent eux-mêmes comme des nœuds F reviennent à *CSMA/CA*. Le nœud récepteur calcule la planification de *TDMA* en fonction des conditions de trafic. Chaque nœud F transmet son paquet programmé à l'intervalle de temps alloué spécifié dans la trame *TDMA*. Afin de permettre la transmission de paquets de données auxquels des slots n'ont pas encore été attribués, une trame *CSMA/CA* est réservée entre deux planifications de trame *TDMA* consécutives comme indiquées dans la figure 2.16.

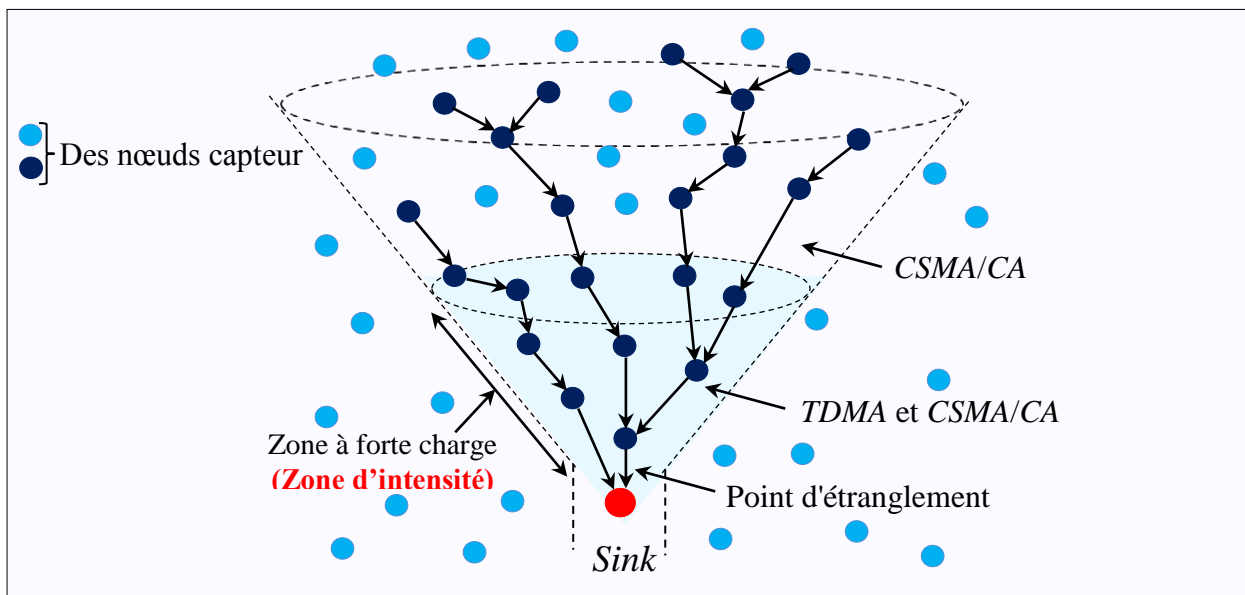


Figure 2.15: L'administration du goulot d'étranglement dans *Funneling-MAC* [74]

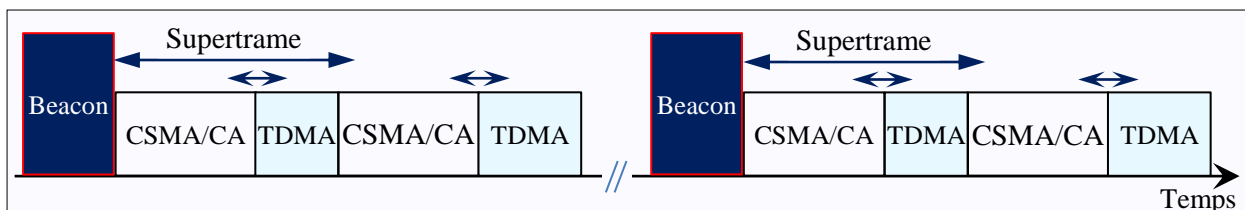


Figure 2.16: Une trame *CSMA/CA* est réservée entre deux planifications de trame *TDMA* dans *Funneling-MAC* [74]

Avec *Funneling-MAC*, les nœuds F recueillent plusieurs balises de beacon et trames de planification en consommant plus d'énergie. Pour éviter que les nœuds qui se trouvent au-delà de la zone de la forte charge interfèrent avec les trames de beacon et des trames programmées qui sont envoyées à une puissance potentiellement élevée. De nombreux

nœuds en permanence dans la région d'intensité risquent de provoquer des interférences en omettant le beacon si leurs heures de réveil périodiques ne correspondent pas à l'heure de beacon. Le *Funneling-MAC* implique également une complexité dans la gestion de la région d'intensité dans le temps et dans l'exécution du CSMA/CA après chaque seconde de la trame TDMA. Le nombre de nœuds F dans un RCSF dense peut être très élevé, ce qui oblige chaque nœud F à attendre plus longtemps avant de pouvoir communiquer avec le récepteur. Cela augmentera la latence du réseau.

2.3.3 La comparaison des différents protocoles MAC pour RCSF

Après une étude approfondie sur les différentes catégories des protocoles MAC discutés ci-dessus, sur leur fonctionnement détaillé, leurs points forts et leurs points faibles, une comparaison claire d'un point de vue purement théorique (sans expérimentations pour des raisons évidentes) présentée dans le tableau 2.1 est effectuée sur la base de plusieurs paramètres tels que : l'efficacité énergétique, le débit, la latence, l'équité, la synchronisation, et la scalabilité.

<i>Protocole</i>	<i>Efficacité énergétique</i>	<i>Latence</i>	<i>Débit</i>	<i>Équité</i>	<i>Synchronisation</i>	<i>Scalabilité</i>
<i>Les protocoles basés sur la contention</i>						
<i>S-MAC</i>	<i>Modérée</i>	<i>Élevée</i>	<i>Faible</i>	<i>Modérée</i>	<i>Oui</i>	<i>Faible</i>
<i>T-MAC</i>	<i>Modérée</i>	<i>Élevée</i>	<i>Faible</i>	<i>Modérée</i>	<i>Oui</i>	<i>Faible</i>
<i>DMAC</i>	<i>Modérée</i>	<i>Élevée</i>	<i>Modéré</i>	<i>Modérée</i>	<i>Oui</i>	<i>Faible</i>
<i>Les protocoles basés sur la planification</i>						
<i>TRAMA</i>	<i>Élevée</i>	<i>Élevée</i>	<i>Élevé</i>	<i>Modérée</i>	<i>Oui</i>	<i>Modérée</i>
<i>FLAMA</i>	<i>Élevée</i>	<i>Élevée</i>	<i>Élevé</i>	<i>Modérée</i>	<i>Oui</i>	<i>Modérée</i>
<i>Les protocoles basés sur l'interrogation du canal</i>						
<i>BMAC</i>	<i>Élevée</i>	<i>Modérée</i>	<i>Modéré</i>	<i>Élevée</i>	<i>Non</i>	<i>Modérée</i>
<i>XMAC</i>	<i>Élevée</i>	<i>Faible</i>	<i>Modéré</i>	<i>Élevée</i>	<i>Non</i>	<i>Modérée</i>
<i>RIMAC</i>	<i>Élevée</i>	<i>Faible</i>	<i>Élevé</i>	<i>Élevée</i>	<i>Non</i>	<i>Élevée</i>
<i>Les protocoles hybrides</i>						
<i>ZMAC</i>	<i>Élevée</i>	<i>Faible</i>	<i>Élevé</i>	<i>Élevée</i>	<i>Non</i>	<i>Modérée</i>
<i>SCP-MAC</i>	<i>Élevée</i>	<i>Faible</i>	<i>Élevé</i>	<i>Élevée</i>	<i>Non</i>	<i>Élevée</i>
<i>Funneling-MAC</i>	<i>Élevée</i>	<i>Faible</i>	<i>Modéré</i>	<i>Modérée</i>	<i>Oui</i>	<i>Élevée</i>

Tableau 2.1: La comparaison des différents protocoles MAC pour les RCSF

Pour conclure cette section, il ressort clairement du tableau 2.1 que la plupart des protocoles discutés ne prennent pas en charge tous les paramètres utilisés dans la comparaison. Ils supportent partiellement un paramètre ou optent pour un autre paramètre, en particulier en faveur de l'énergie. Plusieurs protocoles sont sous-optimaux pour un paramètre donné et peuvent être encore améliorés. De manière significative, la plupart des protocoles considèrent l'efficacité énergétique comme une métrique importante dans leur conception.

2.4 D'autres techniques de conservation d'énergie

Comme cité préalablement ci-dessus dans la section 2.3 de ce chapitre, les mesures expérimentales ont montré que ceux sont généralement les données de transmission qui consomment le plus d'énergie et de manière significative, quand les calculs consomment très peu [75]. La consommation d'énergie du module de détection dépend de la spécificité du capteur. Dans de nombreux cas, il est négligeable par rapport à l'énergie consommée par le module de traitement et, surtout, par le module de communication. Dans d'autres cas, l'énergie utilisée pour la détection peut être comparable ou supérieure à celle requise pour la transmission de données. Ainsi, plusieurs techniques de conservation d'énergie [76] pour les RCSF ont été mise au point, telles que la technique de « *duty-cycling* » détaillée ci-dessus dans la section 2.2 qui permette à un nœud d'éteindre sa radio quand la communication n'est pas nécessaire afin d'économiser plus d'énergie. Dans ce qui suit on présente d'autres techniques de conservation d'énergie citées dans la littérature pour les RCSF, qui sont basés sur les données et la mobilité.

2.4.1 Les techniques dirigées par les données

Les techniques dirigées par les données dans les réseaux de capteurs sans fil visent à diminuer la quantité de données à traiter et à transmettre. Cette technique peut être classée suivant les étapes du traitement des données, en trois catégories: acquisition, traitement et transmission des données.

2.4.1.1 L'acquisition des données

Certaines grandeurs physiques mesurées par les capteurs ne changent pas entre deux échantillons, c'est le cas par exemple d'une mesure de températures où la dynamique est lente. Cela a repoussé les chercheurs à exploiter la corrélation temporelle des données. Les techniques basées sur l'acquisition de données peuvent être classées en deux catégories: les techniques basées sur la prédiction et les techniques basées sur l'échantillonnage.

- **Technique basée sur une prévision de données:** un modèle de prévision est établi pendant les données d'échantillonnage, de sorte que les valeurs futures puissent être prédites avec une certaine précision. Cette approche exploite le modèle obtenu pour réduire le nombre de données acquises, ainsi que la quantité de données à transmettre au *sink*. Un exemple de la technique de prévision centralisée a été proposé dans [77]. Les auteurs ont proposé un modèle de prévision pour les applications de surveillance de l'environnement appelé « *PREMON* » qui repose sur deux principes. La première consiste à exploiter les corrélations spatiales. Reconnaisant les efforts d'un capteur, il devient inutile pour un voisin de retransmettre tous les paquets reçus, mais uniquement les modifications pertinentes. La seconde est basée sur le fait qu'un aperçu du réseau peut être considéré comme une image et que, par conséquent, l'évolution des mesures peut être visualisée sous forme de vidéo.

▪ **Technique basée sur l'échantillonnage:** les techniques basées sur l'échantillonnage peuvent être classées en trois catégories: l'échantillonnage adaptatif, l'échantillonnage hiérarchique et l'échantillonnage actif basé sur un modèle. Comme les échantillons mesurés peuvent être corrélés, les techniques d'échantillonnage adaptatif exploitent ces similitudes pour réduire le nombre d'acquisitions. L'approche d'échantillonnage hiérarchique suppose que les nœuds sont équipés de capteurs (ou détecteurs) de différents types. Chaque capteur étant caractérisé par une résolution donnée et la consommation d'énergie associée, cette technique sélectionne dynamiquement la classe à transformer pour obtenir un compromis entre précision et économie d'énergie. L'échantillonnage actif basé sur un modèle adopte une approche similaire pour la prévision des données. Un modèle du phénomène mesuré est établi pendant les données d'échantillonnage, de sorte que les valeurs futures peuvent être prédites avec une certaine précision. Cette approche exploite le modèle résultant pour réduire le nombre d'échantillons de données, ainsi que la quantité de données à transmettre au *sink*, bien que ce ne soit pas leur objectif principal.

2.4.1.2 Traitement et transmission des données

Plusieurs approches de traitement de données ont été proposées dans la littérature pour résoudre le problème de la conservation de l'énergie dans les capteurs sans fil. Nous présentons les deux principaux. Le premier est basé sur la compression de données et le second sur l'agrégation de données [78].

1. Compression des données

L'avantage des techniques de compression est de permettre à moins de données de communiquer et d'économiser de l'énergie, ce qui entraîne un allongement de la durée de vie des réseaux. Cependant, les algorithmes de compression couramment utilisés sur les ordinateurs (tels que *Lempel-Ziv*, *WinZip* ou *JPEG*) ne sont pas directement transférables à tous les capteurs, car beaucoup supposent trop de calculs par rapport au gain fourni. Il existe plusieurs méthodes de compression de données. Les principales méthodes de compression adaptées aux données de réseaux de capteurs sans fil sont les suivantes:

▪ **Le codage par planification:** la compression dans le codage par planification consiste à supprimer les informations redondantes des capteurs (adresse de destination, erreurs de code de contrôle, synchronisation d'horloge) et à fusionner les données restantes. Ainsi, le nombre de paquets envoyés est réduit. Lors du codage par planification, les nœuds envoient leurs paquets à un nœud unique appelé « *nœud de compression* ». Le paquet codé est construit à partir de combinaisons des paquets reçus de plusieurs nœuds. Le protocole " *Funning* " [79], basé sur l'approche par programmation, réduit considérablement la quantité d'énergie consommée lors de la communication, avec une économie d'énergie pouvant atteindre 44 %. Cet objectif peut être atteint en n'envoyant qu'un seul flux de données à un groupe de capteurs au lieu que chaque capteur individuel envoie ses données.

- **La compression distribuée [80]:** chaque capteur fournit une représentation binaire de ses échantillons en tenant compte de la corrélation avec les échantillons mesurés par les autres capteurs. La communication entre capteurs est interdite, ce qui rend difficile l'estimation de cette corrélation dans la pratique.

2. L'agrégation des données

Les RCSF sont généralement assez denses, cela signifie que les nœuds suffisamment proches (voisins) peuvent capturer les mêmes données (température, pression, équivalent d'humidité, par exemple). Par conséquent, plusieurs études ont été menées pour éliminer la redondance et réduire le trafic de données sur le réseau. Le mécanisme consiste à traiter les données collectées par chaque capteur dans un nœud appelé « *nœud agrégateur* ». Seul le résultat obtenu sera transmis au *sink*. De cette manière, la quantité de données communiquées sur le réseau peut être réduite, ce qui réduit par conséquent la consommation de bande passante et l'épuisement de l'énergie des capteurs. Les techniques d'agrégation de données peuvent être divisées en deux:

- **L'agrégation centralisée:** agrégation dans les clusters, formée via un protocole de clustering. Les premières zones (*clusters*) sont définies via un protocole, puis les données sont agrégées dans ces zones via un cluster-head. Ces derniers peuvent éventuellement évoluer dans le temps afin de mieux répartir la consommation d'énergie entre tous les nœuds du réseau.

- **L'agrégation distribuée:** l'agrégation est répartie dans un arbre d'agrégation, c'est-à-dire que le réseau est considéré comme un tout. Le protocole COUGAR [81] est un exemple d'agrégation de données distribuées. Les données produites par le réseau sont modélisées comme une table relationnelle, le réseau est vu comme une grande base de données distribuée. Les attributs de cette table sont des informations de capteur ou des données produites par le capteur car COUGAR fournit une agrégation partielle au niveau des nœuds.

2.4.2 Les techniques basées sur la mobilité

Dans certains cas, lorsque les nœuds sont mobiles, la mobilité peut être utilisée comme un outil permettant de réduire la consommation d'énergie. Dans un réseau statique de capteurs, les paquets provenant des nœuds suivent les chemins multi-acheminements vers le *sink*. Ainsi, certains chemins peuvent être chargés, et les nœuds situés à proximité de *sink* relaient davantage les paquets et sont plus sujets à un épuisement prématuré de leurs batteries. Si certains nœuds sont mobiles, le trafic peut être modifié si les nœuds mobiles sont chargés de collecter les données directement à partir de nœuds statiques. Les nœuds ordinaires attendent le passage d'une unité mobile pour envoyer leurs messages afin que la communication ait lieu à proximité. Par conséquent, les nœuds ordinaires peuvent économiser de l'énergie car la longueur du chemin, les collisions et les messages de contrôles (*overhead*) de diffusion sont ainsi réduits. En outre, le périphérique mobile peut visiter le réseau pour répartir uniformément la consommation d'énergie due à la communication. Lorsque le coût de la mobilité des

nœuds de capteur est excessif, l'approche traditionnelle consiste à attacher un capteur à des entités qui se déplacent dans le champ de détection, telles que des bus ou des animaux. Les techniques basées sur la mobilité peuvent être classées en deux groupes:

Les techniques avec un *sink* mobile et les techniques avec des nœuds relais mobiles, en fonction du type d'entité mobile. Il est important de noter ici que lorsque nous examinons les systèmes mobiles, un problème important est le type de contrôle de la mobilité des nœuds conçu par *Integra*, décrit dans [82]. Les nœuds mobiles peuvent être divisés en deux catégories: ils peuvent être spécifiquement conçus en tant que partie de l'infrastructure de réseau ou de l'environnement. Lorsqu'ils font partie de l'infrastructure, la mobilité peut être entièrement contrôlée dans la mesure où ils sont généralement robotisés. Lorsque les nœuds mobiles font partie de l'environnement, ils peuvent ne pas être contrôlables. S'ils suivent un horaire strict, ils ont une mobilité totalement prévisible (par exemple, une navette pour les transports en commun). Sinon, ils peuvent avoir un comportement aléatoire tel qu'aucune hypothèse ne peut être faite sur leur mobilité. Enfin, ils peuvent suivre un schéma de mouvement qui n'est ni prévisible ni totalement aléatoire. Par exemple, dans le cas d'un bus se déplaçant dans une ville dont le tarif est soumis à des variations importantes en raison des conditions de circulation. Dans ce cas, les schémas de mobilité peuvent être établis à partir d'observations et d'estimations d'une certaine précision.

2.5 Conclusion

Les *RCSF* attirent beaucoup de chercheurs en raison de leurs applications réellement utiles dans presque tous les domaines. Ainsi, que la conception d'un protocole *MAC* susceptible d'améliorer l'efficacité énergétique et de prolonger la durée de vie du *RCSF* est un problème complexe. Dans ce troisième chapitre, nous avons présenté un état de l'art sur les travaux recensés dans la littérature, qui concernent principalement la conservation de l'énergie dans un *RCSF*. Nous avons d'abord présenté la sous couche *MAC* et la technique « *duty-cycling* » de la conservation d'énergie la plus utilisée dans les *RCSF*. À cette fin, nous avons exposé le fonctionnement détaillé, les points forts, les points faibles et la classification de plusieurs protocoles *MAC* développés pour les *RCSF* fonctionnant avec un faible cycle d'activité (*duty-cycle*). Par la suite nous avons donné une comparaison d'un point de vue théorique pour différentes métriques dans le tableau 2.1 de ces protocoles *MAC* débattus, ce qui montre que la majorité de ces protocoles ont une efficacité énergétique. À la fin de ce chapitre nous avons présenté brièvement d'autres techniques de conservation d'énergie citées dans la littérature qui s'intéressent à l'acquisition, au traitement et à la transmission de données, ainsi qu'à des techniques utilisent la méthode de la mobilité des nœuds. Il existe bien sûr de nombreuses autres techniques de conservation de l'énergie. Par exemple, à partir de paradigmes de systèmes auto-organisés et l'inter couches. Dans le chapitre suivant, nous exposerons la théorie de la récolte énergétique et un état de l'art des protocoles *MAC* particulièrement proposés pour les *RCSF* à récupération d'énergie.

Les protocoles MAC pour les RCSF à récupération d'énergie

3.1 Introduction

La technologie de récupération d'énergie (*RE*) dans les réseaux de capteurs sans fil gagne en popularité en supprimant la charge de remplacer/recharger les sources d'énergie épuisées par des dispositifs de récupération d'énergie. *RE* fournit une source d'énergie alternative à partir de l'environnement, par conséquent, en exploitant le processus de *RE*, le *RCSF* peut atteindre une vie permanente. À la lumière de cela, l'accent est mis sur la conception de nouveaux protocoles *MAC* qui visent à maximiser la durée de vie des réseaux en utilisant autant d'énergie que possible plutôt que de laisser le nœud finir toute son énergie et mourir, puisque l'énergie récoltée est plus grande que celle consommée, si nous savons la gérer. Plusieurs protocoles *MAC* ont été proposés pour exploiter l'énergie environnante afin d'alimenter les réseaux de capteurs sans fil à récupération d'énergie (*RCSF-RE*). Dans ce chapitre, les caractéristiques de base de la structure de *RCSF-RE* ont été clarifiées. Par la suite, plusieurs protocoles *MAC* proposés pour le *RCSF-RE* sont présentés, ainsi qu'une description de leurs principes de fonctionnement et de leurs caractéristiques de base et à la fin une comparaison théorique est faite entre ces protocoles.

3.2 Définition de la récupération énergétique

La récupération d'énergie (« *energy harvesting* » ou « *energy scavenging* » en anglais) est définie comme la transformation de l'énergie naturelle ambiante, non électrique et renouvelable présente dans l'environnement externe en une énergie électrique, directement utilisable ou stockée dans des unités comme des batteries afin de servir au fonctionnement d'appareils. Néanmoins, le principal avantage de la récupération d'énergie est d'éviter l'obligation de la charge fréquente pour remplacer ou recharger les batteries des dispositifs et aussi d'augmenter la durée de vie et répondre aux problématiques actuelles d'écologie [Web 15].

3.3 Les nœuds de capteur à récupération d'énergie

La récupération d'énergie se réfère à la récolte d'énergie ou à la conversion d'énergie d'une forme à l'autre. Appliquée aux nœuds de capteurs, l'énergie provenant de sources externes peut être utilisée pour alimenter les nœuds capteur, ce qui augmente leur durée de vie et leurs capacités de fonctionnement.

3.3.1 Architecture du système de récupération d'énergie

En plus les composants du nœud capteur mentionnés ci-dessus dans le chapitre 1, section 1.3, chaque nœud capteur est doté d'un système de récupération et de transformation d'énergie en énergie électrique qui peut se décliner en deux architectures différentes avec ou sans capacité de stockage [15]: (a) *récupérer-dépenser*: l'énergie récoltée est utilisée directement et (b) *récupérer-recharger-dépenser*: l'énergie récoltée est stockée pour une utilisation ultérieure.

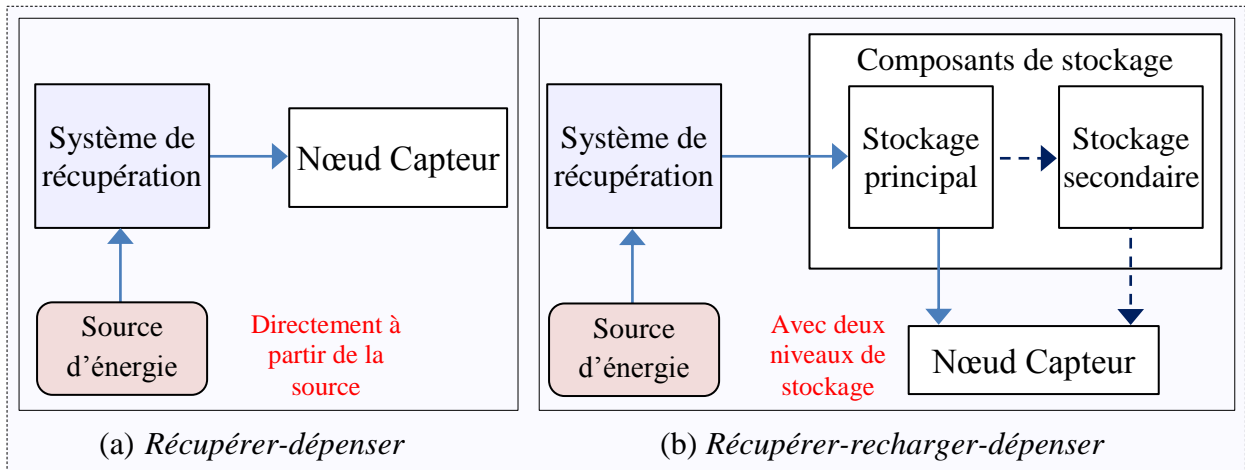


Figure 3.1: Les deux architectures d'un système de récupération d'énergie [15]

3.3.1.1 Architecture *Récupérer-dépenser*

Dans la première architecture du système de récupération (*Récupérer-dépenser*) illustré en figure 3.1(a), le nœud capteur est alimenté directement par le système de récupération qui est actif tant que l'énergie récupérée dépasse le minimum d'énergie nécessaire pour son fonctionnement sinon si ce n'est pas le cas alors le nœud capteur sera désactivé. En conséquence, l'activation ou la désactivation dans ce cas du nœud capteur dépend des changements brusques de la quantité d'énergie récoltée par rapport à la capacité minimale imposée. Des exemples de sources d'énergie utilisées pour la conception d'un système basé sur cette structure telles que: des sources mécaniques comme; la marche à pied et la poussée des boutons ou des touches qui sont basées sur la déformation d'un matériau piézoélectrique qui sont placés stratégiquement dans des chaussures et dans la structure des boutons et des touches respectivement pour engendrer de l'énergie électrique soit en marchant ou en courant et par l'envoi d'un court message sans fil respectivement.

3.3.1.2 Architecture *Récupérer-recharger-dépenser*

Dans la seconde architecture du système de récupération (*Récupérer-recharger-dépenser*) illustré en figure 3.1(b), l'énergie récoltée est stockée à préalable avant toute exploitation possible, cette architecture comprend un composant de stockage principal qui stocke l'énergie récoltée et qui peut être dupliqué pour servir comme un composant de

stockage secondaire backup en cas de problème d'épuisement où les deux composants servent comme une source d'alimentation pour le nœud capteur. La caractéristique importante qui distingue cette architecture est que tout surplus d'énergie sera conservé et utilisé plus tard lorsque la possibilité de récupération est non disponible ou que la consommation d'énergie du nœud du capteur doit être augmentée pour améliorer ses performances. Les systèmes de récupération d'énergie solaire sont parmi les exemples de systèmes qui utilisent cette architecture lorsque la source d'énergie disponible est prévisible mais non contrôlable. Un nœud de capteur dans la journée utilise cette énergie pour fonctionner et le stocke pour une utilisation ultérieure et pendant la nuit, utilise l'énergie stockée pour qu'il soit opérationnel.

3.4 Les sources d'énergie pour la récupération énergétique

Représente la source d'énergie pour toute architecture de récupération d'énergie, l'élément de base qui lui permet de connaître la quantité et le taux d'énergie disponible pour l'utilisation. Cependant, plusieurs sources d'énergie exploitées possèdent des caractéristiques différentes et peuvent être contrôlables ou non contrôlables [22]. Dans les sources d'énergie contrôlables, la disponibilité de l'énergie ne peut être prédite avant la récolte mais elles fournissent de l'énergie requise si nécessaire. D'autre part, les sources d'énergie non contrôlables permettent de récupérer l'énergie seulement si elle est disponible pendant des périodes différentes. Dans ce cas, un modèle de prédiction de l'énergie qui prévoit la disponibilité de la source d'énergie est indispensable. Si le modèle de prédiction est difficile on parle de source non contrôlable et imprévisible. De manière générale, les sources énergétiques peuvent être séparées en deux catégories [85]: ambiante et humaine. Dans les sources d'énergie ambiante l'énergie est récoltée à partir de l'environnement externe telles que, l'énergie solaire, l'énergie de vibration, l'énergie éolienne et l'énergie de la radiofréquence (*RF*) et dans les sources d'énergie humaines de la deuxième catégorie l'énergie est récoltée par les mouvements corporels qui peuvent être passive non contrôlable par l'utilisateur comme, la respiration et la chaleur corporelle ou active contrôlable par l'utilisateur comme, la marche et les mouvements des doigts.

3.5 Les techniques de la récupération énergétique

La figure 3.2, montre quelques exemples de techniques de récupération d'énergie fréquemment utilisées et qui seront décrits brièvement par la suite dans cette section.

3.5.1 Récupération de l'énergie mécanique

La récupération de l'énergie mécanique désigne la transformation de l'énergie mécanique en une énergie électrique en utilisant la pression, les vibrations, les moteurs à charge élevée et la force [86]. Ce type d'énergie ambiante converti représente une technique attirante pour l'alimentation des nœuds de capteurs sans fil et qui peut exister presque dans tous les endroits de déploiement des *RCSF*. Comme exemple, le mécanisme

piézoélectrique bénéficie des propriétés de certains matériaux dont les cristaux se polarisent électriquement lorsqu'ils sont soumis à une force mécanique. La chaussure engendrant de l'énergie pendant la marche et qui a été développée par l'équipe MIT (*Media laboratory*) représente le modèle le plus remarquable qui transforme l'énergie mécanique en une énergie électrique.

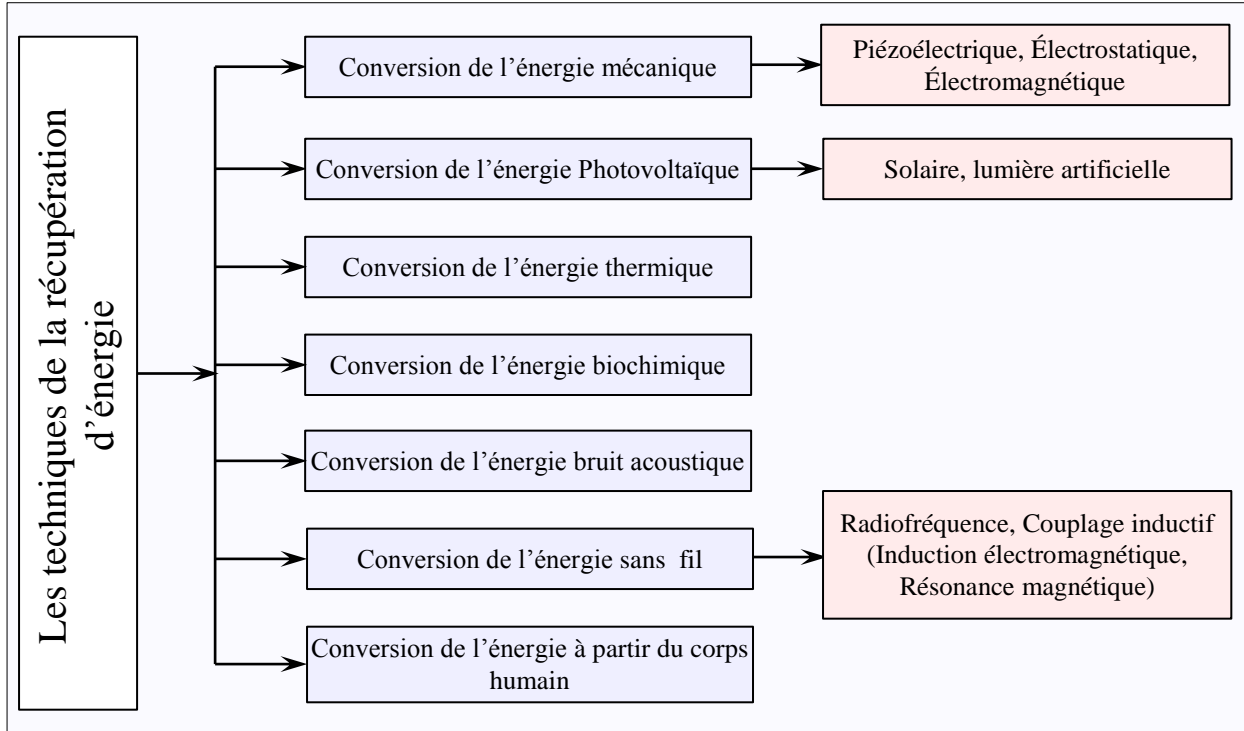


Figure 3.2: quelques exemples de techniques de récupération d'énergie

3.5.2 Récupération de l'énergie photovoltaïque

Cette technique de conversion de ce type d'énergie dépend des conditions environnementales disponibles ainsi que les moyens utilisés qui peuvent influencer sur l'efficacité du système de la récolte utilisée et sur la quantité d'énergie récoltée [87]. La récupération de l'énergie photovoltaïque se réfère à la transformation des photons provenant de la lumière solaire, artificielle ou d'autres en énergie électrique sur la base des cellules lumières et qui peut engendrer des niveaux plus élevés d'électricité par comparaison à d'autres techniques. Ainsi, ces sources conviennent aux systèmes de récolte d'énergie à grande échelle [86].

3.5.3 Récupération de l'énergie thermique

La récupération d'énergie thermoélectrique consiste fréquemment en l'emploi de générateurs thermoélectriques [88] pour la génération de l'énergie électrique à partir de la différence de températures (gradients thermiques) qui maintient la récolte de l'énergie en permanence [128]. Ainsi, elle offre des degrés hauts d'énergie récoltés. Le récolteur d'énergie récolte l'énergie électrique quand une tension électrique est engendrée par deux métaux joints à deux jonctions qui sont maintenues à des températures différentes. En fait, les métaux interagissent différemment avec la variation de température, entraînant un

flux de chaleur à travers le générateur thermoélectrique. Cela engendre une différence de tension proportionnelle à la différence de température entre les plaques chaudes et froides.

3.5.4 Récupération de l'énergie biochimique

Le processus de transformation de l'oxygène et des substances chimiques en électricité par le biais de réactions électrochimiques désigne la récupération de l'énergie biochimique. En particulier, l'énergie biochimique peut être convertie en énergie électrique en utilisant des cellules de biocarburant qui agissent comme des enzymes et des catalyseurs actifs. Parmi ceux-ci, le glucose est considéré comme la source la plus commune des combustibles utilisés. Le rendement de ce type de récupération d'énergie repose sur le type et de la disponibilité des piles à combustible quoiqu'elle puisse être excellente à d'autres techniques de récupération d'énergie en matière de puissance de sortie continue [128].

3.5.5 Récupération de l'énergie acoustique

Le processus de conversion des ondes sonores de l'environnement en énergie électrique à l'aide d'un résonateur indique la récolte de l'énergie acoustique. Cette dernière est exploitable dans des endroits isolés ou éloignés, ou des systèmes fermés où le câblage et les commutations électriques sont difficiles à utiliser. Bien que, la productivité de la puissance acoustique récupérée est faible et une telle énergie ne peut être récupérée que dans des environnements très résonnants [128].

3.5.6 Récupération de l'énergie des signaux radio fréquence (RF)

Avec l'augmentation de la prolifération des émetteurs radios (télévision, radio, téléphonie, ...), en particulier en milieu urbain, les émissions radioélectriques peuvent être considérées comme des sources d'énergie ambiante. Ainsi, il est possible de récolter l'énergie à partir de ces signaux radio fréquence (RF) qui se propagent dans l'air. Par exemple, à partir des signaux *Wifi*, les signaux de signalisation dans les réseaux cellulaires, etc. Cette technique de récolte d'énergie requiert une densité suffisante de signaux RF pour être effective. Cependant, la principale limitation de la récupération d'énergie RF reste la nécessité de se placer dans le champ proche de la source ou d'avoir une surface d'antenne très importante afin d'avoir une puissance récupérée significative.

3.5.7 Récupération de l'énergie à partir du corps humain

Selon [88], le corps humain brûle en moyenne d'environ 10,5 mégajoules d'énergie par jour, ce qui équivaut à la dissipation de puissance moyenne de 121 watts. Ainsi, Le corps humain présente une source attirante pour la génération d'énergies à partir de la dissipation de chaleur, des rotations, l'application du poids du corps ainsi que la déformation élastique des tissus et autres accessoires. En effet, cela donne des possibilités de récupération d'énergie pour faire fonctionner des dispositifs médicaux implantables ou mobiles pouvant être utilisés pendant une longue période ou rechargés en permanence.

Selon [Web 17], plusieurs appareils de mouvement humain sont utilisés pour la récupération d'énergie.

3.6 Quelques applications des nœuds de capteur à récupération d'énergie

Dans cette section, nous présentons des implémentations de *RCSF-RE* pour les applications environnementales suivantes: suivi des animaux, surveillance de la qualité de l'air (*AQM : Air Quality Monitoring*), surveillance de la qualité de l'eau (*WQM : Water Quality Monitoring*) et surveillance des catastrophes (*DM : Disaster Monitoring*). Dans ces applications environnementales, différents protocoles spécifiques aux applications sont conçus pour répondre aux critères définis pour que le système atteigne des performances élevées.

3.6.1 Suivi des animaux

Les *RCSF-RE* sont utilisés pour le suivi des animaux afin de fournir des informations sur différentes espèces d'animaux. Les informations rassemblées dans ces systèmes comprennent des données sur les habitudes, les comportements et les conditions de l'habitat des animaux; ces informations peuvent aider les chercheurs à mieux comprendre le statut et la dynamique des ressources fauniques et peuvent donc constituer la base d'une protection efficace, d'une utilisation durable et d'une gestion scientifique des ressources fauniques [89]. Quelques exemples de projets d'applications qui ont été réalisés dans ce domaine sont énoncés ci-dessous:

3.6.1.1 *ZebraNet*

ZebraNet [90] est un projet mené par l'Université de Princeton et déployé dans le centre du Kenya. Il vise à suivre le mouvement des zèbres en utilisant des colliers capteurs fixés sur ces animaux. Ils ne pèsent que quelques centaines de grammes et produisent 0,4W en plein soleil [90]. Le collier *ZebraNet* dispose de 14 modules solaires (chacun doté de 3 cellules solaires en série), d'un simple comparateur et d'un convertisseur élévateur. Chaque module solaire produit un maximum de 7 mA à 5V. Les sorties des modules solaires sont connectées ensemble en parallèle, ce qui entraîne l'ajout de la puissance générée par chacun d'eux. Par conséquent, le *ZebraNet* est défini comme un réseau de capteurs mobile avec une couverture réseau réduite et des capteurs à haute énergie contenant une mémoire flash, un petit processeur (*CPU*), un système de localisation (*GPS*) et un transducteur sans fil.

3.6.1.2 *TurtleNet*

TurtleNet [Web 18] est un projet dont le but est de résoudre les problèmes de détection et de communication liés au suivi des sites des tortues. Il s'agit d'un effort similaire au projet *ZebraNet* et s'appuie sur la contribution de *ZebraNet* à l'alimentation de capteurs portables utilisant des cellules solaires, pour effectuer un suivi perpétuel de la faune. Comme on s'attend à ce que les tortues passent une grande partie de leur temps

sous l'eau, le nœud est imperméabilisé en le conditionnant dans des tubes rétractables et en scellant les extrémités avec un époxyde résistant à l'eau. Le nœud *TurtleNet eFlux* utilise une batterie rechargeable Li-ion chargée à l'aide d'une cellule solaire. Le module de charge et d'énergie peut gérer une grande variété de cellules solaires [Web 18].

3.6.2 La surveillance de la qualité de l'air

L'utilisation des *RCSF-RE* dans la surveillance de la qualité de l'air contribue à la détection de polluants atmosphériques tels que les particules, le dioxyde de carbone (CO_2), l'ozone (O_3), le dioxyde d'azote (NO_2) et le dioxyde de soufre (SO_2).) [91] [92] [93]. Les *RCSF-RE* pour la surveillance de la qualité de l'air peuvent être installés à l'intérieur ou à l'extérieur, la principale différence étant la quantité d'énergie récupérée dans l'environnement. Pour les systèmes *AQM* extérieurs, les cellules solaires sont une technique de récupération d'énergie évidente, mais elles doivent être accompagnées de stockage, de gestion et de prévision d'énergie afin de fournir un rendement élevé et de surmonter les obstacles à la récupération de l'énergie solaire, tels que les conditions météorologiques. Les auteurs dans [94] proposent une mise en œuvre de la récupération d'énergie hybride thermique solaire pour la surveillance de la qualité de l'air (2015). Les auteurs ont étudié la possibilité de récolter de l'énergie à partir de déchets d'usines d'engrais organiques afin de surveiller la pollution de l'air en termes de concentration de gaz ammoniac. La première partie du système est le récupérateur d'énergie. Un générateur thermoélectrique (*TEG*) est utilisé pour convertir la différence entre les côtés chaud et froid d'un réservoir de désodorisant à gaz - utilisé pour traiter l'air pollué - de l'énergie thermique à l'énergie électrique. Afin d'augmenter la puissance nécessaire au fonctionnement du capteur d'ammoniac et du module de transmission le matin, des cellules solaires sont proposées.

3.6.3 La surveillance de la qualité de l'eau

La surveillance de la qualité de l'eau reste un domaine d'application intéressant où les *RCSF-RE* peuvent être utilisés pour surveiller différentes conditions afin d'assurer de l'eau potable saine. Les *RCSF-RE* qui récupèrent l'énergie de leur environnement peuvent permettre de surveiller la qualité de l'eau dans les zones rurales et résidentielles. L'énergie peut être récupérée à partir de sources solaires, thermiques et mécaniques pour prendre en charge ce type d'application environnementale, qui nécessite une énergie constante pour mesurer des paramètres tels que la température, le *pH*, la conductivité, la turbidité et l'oxygène dissous. Outre la surveillance de l'eau potable pour l'homme, une surveillance de la qualité de l'eau peut être effectuée pour améliorer la vie dans les environnements aquatiques. L'auteur dans [95] fournit un exemple du système de gestion de la qualité totale avec implémentation de récupération d'énergie. Dans leurs travaux, les auteurs ont proposé un système de récupération d'énergie pour la surveillance et la gestion de l'aquaculture (surveillance de la qualité des eaux de pêche) afin de collecter, traiter et transmettre les paramètres environnementaux de la pisciculture. Le *RCSF* a été conçu

pour détecter la température de l'eau, le pH , l'oxygène dissous, la conductivité électrique et le niveau d'eau. Des capteurs *WQM*, tels que le capteur de pH et le capteur de turbidité, ont été installés pour détecter un phénomène spécifique, et les données collectées ont été transmises à une station *WQM* distante pour le traitement de données.

3.6.4 La surveillance de la santé

Les nœuds de capteurs sans fil peuvent être déployés en tant que réseau *BAN* pour les soins préventifs, afin de surveiller les fonctions et les attributs physiques du corps humain, ce qui peut améliorer la santé d'un individu grâce à une surveillance continue et à une réduction des coûts de cette surveillance [96] [97]. La récupération d'énergie provenant de diverses sources, telles que les sources biochimiques et biomécaniques du corps humain, en plus de récupérer l'énergie ambiante, permet aux réseaux *BAN* d'être plus pratiques en réduisant la taille et le coût de leur système [98]. Le corps humain gaspille environ 100 *W* de chaleur, pouvant être récupérée par des générateurs thermoélectriques (*TEG*) et utilisée pour alimenter des réseaux *BAN* [99]. À titre d'exemple, une étude expérimentale des propriétés thermiques humaines du corps humain avec un collecteur thermoélectrique attaché (*TEH*) est réalisée dans [100] et montre que le dispositif *TEH* est capable d'alimenter des dispositifs de surveillance de l'état de santé de faible puissance tels que; l'électrocardiographie sans fil par récupération de la chaleur corporelle. Les transmissions *RF* provenant de télécommunications commerciales, en particulier du système cellulaire 900/188 *MHz*, représentent une énergie ambiante fiable pouvant être exploitée pour alimenter des capteurs biomédicaux portables [101]. Par exemple, dans [102], un capteur sans pile avec récupérateur d'énergie *RF* à suivi d'efficacité et antenne fouet a été proposé. Des expériences pratiques montrent que le système peut mesurer et transmettre avec succès la température abdominale à une station de base sur une distance maximale de 9 *m*.

3.6.5 La surveillance des catastrophes

Les catastrophes naturelles telles que les tremblements de terre et les inondations ont de graves répercussions sur l'humanité en termes de coût de la vie, de coûts financiers et de troubles émotionnels [103]. Prévenir, gérer ou au moins atténuer leurs effets est un défi majeur en raison de la nature imprévisible, de courte durée et a fort impact de ces catastrophes. Cela a conduit à des recherches axées sur l'utilisation de réseaux de récupération d'énergie basés sur les vibrations pour surveiller ou même gérer les effets de catastrophes naturelles [104] [105]. Un système de surveillance des inondations qui récupère l'énergie d'un panneau solaire 2 *V*, 150 *mA* et qui charge un super-condensateur de 0,47 *F* est mis en œuvre dans [105]. Un circuit intégré de gestion de l'alimentation *BQ25570* est utilisé comme conditionneur d'alimentation avec un microcontrôleur *MSP430G2553* [Web 19] de Texas Instruments à faible consommation d'énergie. Les résultats montrent qu'un maximum de 0,3 *W* peut être récolté avec cette configuration. La conception du réseau était basée sur le *clustering*. Trois nœuds de capteurs principaux ont

été utilisés dans la configuration en grappes (à savoir, nœuds de capteurs de vibrations, de températures et acoustiques) [103]. Dans chaque grappe, les têtes de grappe et la station de base ont plus de puissance et d'énergie de calcul. Le protocole *MAC EDEH* (*Event-Driven Energy-Harvesting*) a été utilisé pour résoudre les conflits de canaux sous des contraintes d'énergie strictes. Le protocole utilisait un exposant de *back-off* important et étalait l'accès des périphériques sur une période plus longue, ce qui réduit les risques de collision au prix d'un retard accru.

3.7 Les motivations pour les protocoles MAC spéciaux pour les RCSF-RE

Les protocoles *MAC* simples ne pouvant pas satisfaire aux exigences imposées par les *RCSF-RE*, il est inévitable de prendre en compte des protocoles *MAC* spéciaux personnalisés pour les *RCSF-RE*. Pour acquérir une compréhension approfondie des problèmes associés aux protocoles simples, il est important de comprendre en quoi les *RCSF* simples sont différents des *RCSF-RE* et pourquoi les protocoles *MAC* simples se comportent de manière inappropriée pour *RCSF-RE*. Les protocoles *MAC* déjà disponibles dans la littérature pour les réseaux non-récupérateurs d'énergie ont pour objectif de prolonger la durée de vie du réseau en évitant les opérations à énergie exhaustive énumérées à la Section 1.3.7 du chapitre 1. D'autre part, les protocoles *MAC* spéciaux pour *RCSF-RE* visent à réaliser le meilleur compromis entre des conditions d'énergie incertaines et une durée de vie du réseau plus longue avec des performances optimales. Plusieurs facteurs expliquent le besoin d'un type spécial de protocole *MAC* destiné à *RCSF-RE* et présentés dans cette section.

3.7.1 Le principe de conception d'un RCSF-RE

Il existe une différence fondamentale dans le principe de conception d'un *RCSF* à récupération d'énergie (*RCSF-RE*) par rapport à un *RCSF* fonctionnant sur batterie puisque ce dernier a été développé dans le but de prolonger leur durée de vie. Au contraire, le paradigme de la récupération d'énergie atténue les contraintes de puissance auxquelles sont confrontés les *RCSF* fonctionnant sur la batterie. Un *RCSF-RE* s'attache plutôt à améliorer les performances du réseau (débit, délai, le temps inter-arrivé, etc.) fonctionnant dans un état énergétique durable. Par conséquent, le principe de conception différencie le besoin d'un protocole *MAC* spécial qui devrait être conçu en tenant compte des exigences de performances (c'est-à-dire ciblant l'amélioration de la qualité de service au lieu d'une durée de vie plus longue) d'un *RCSF-RE*.

3.7.2 L'adaptabilité de la technique duty-cycling

Les nœuds individuels dans un *RCSF* ordinaire subissent généralement un cycle d'activité (*duty-cycle*) commun en raison de la disponibilité évidente de l'énergie car ils sont équipés d'une batterie et décroissent progressivement avec le temps. Dans *RCSF-RE*, la quantité d'énergie réellement disponible dans un cas donné n'est pas claire en

raison de certaines limitations environnementales des mécanismes de récolte. Par conséquent, la conception du protocole *MAC* pour *RCSF-RE* exige un duty-cycle adaptatif de nœuds individuels par rapport à un duty-cycle commun à l'ensemble du système, basé sur leur disponibilité individuelle en énergie. Ce type particulier de protocole *MAC* permettrait aux nœuds à faible énergie de gérer leurs opérations (commutation entre les modes veille et réveil) indépendamment du système global, induisant une flexibilité de fonctionnement.

3.7.3 Les capacités de la récolte

Contrairement aux nœuds du *RCSF* standard, les nœuds d'un *RCSF-RE* sont équipés d'un système de récupération d'énergie qui leur permet de récupérer une certaine quantité d'énergie de l'environnement. D'une part, cela aide ces nœuds à poursuivre leurs opérations en cours mais, d'autre part, cela peut poser un problème pour les nœuds, car les fonctionnalités de collecte ne sont pas les mêmes pour tous les nœuds d'un *RCSF-RE*. Cette variation peut être due à plusieurs facteurs, notamment le mécanisme de récolte, le temps, l'environnement ou la position précise des nœuds. Par exemple, la quantité d'énergie récupérée par un nœud équipé de cellules solaires positionnées dans des rayons directs du soleil serait différente de celle d'un nœud placé par hasard dans une zone ombragée. Le protocole *MAC* spécial doit être conçu pour compenser intelligemment les nœuds d'énergie inférieure, ce qui permet de récupérer le surplus d'énergie récupéré par les nœuds dotés de capacités supérieures.

3.7.4 Fonctionnement neutre en énergie

Un fonctionnement neutre en énergie (*Energy Neutral Operation (ENO)* en anglais) constitue un défi essentiel pour équilibrer la consommation d'énergie et optimiser les performances en fonction des niveaux d'énergie actuels et prévus. La maximisation de la durée de vie d'un *RCSF-RE* implique la maximisation des performances d'un nœud tout en s'assurant que sa batterie ne s'épuise pas avant le prochain cycle de recharge. Un nœud prend en compte les niveaux d'énergie actuels et attendus, règle dynamiquement les performances et garantit simultanément que le nœud ne fonctionne pas en dessous des niveaux de performance minimums et ne bascule pas avant le prochain cycle de recharge. Le fonctionnement neutre au niveau des nœuds est le fonctionnement perpétuel d'un nœud de capteur, c'est-à-dire que la consommation d'énergie d'un nœud est toujours inférieure à l'énergie récoltée. Sur la base du type d'architecture de récupération d'énergie utilisé, le fonctionnement neutre au niveau des nœuds a différentes implications.

Soit $P_S(t)$ la puissance engendrée par une source d'énergie à l'instant t et $P_C(t)$ la puissance consommée par le nœud du capteur. La condition pour un fonctionnement neutre en énergie [15][22] pour les deux architectures de récupération d'énergie citée ci-dessus (voir la section 3.3) est la suivante :

a- Dans l'architecture Récupérer-dépenser : dans ce cas [15], l'énergie récupérée est utilisée directement et d'une façon continue par le nœud du capteur. Deux cas peuvent être distingués pour définir le fonctionnement du nœud, ces cas sont :

- 1- Si $P_S(t) \geq P_C(t) \forall t$ (l'énergie récoltée est supérieure à celle consommée par le nœud), alors le nœud est en état de fonctionnement neutre en énergie sinon si $P_S(t) = P_C(t) \forall t$ (l'énergie récoltée est égale à celle consommée par le nœud), alors le nœud est en état de fonctionnement neutre en énergie maximale (*ENOMAX*). Comme le montre la figure 3.3(a) où $P_S(t) - P_C(t)$ est l'énergie gaspillée ;
- 2- Si $P_S(t) < P_C(t) \forall t$ (l'énergie récoltée est inférieure à celle consommée par le nœud), alors le nœud ne fonctionne pas;

b- Dans l'architecture Récupérer-recharger-dépenser : dans ce cas [15], l'énergie récupérée est stockée à l'avance dans l'unité de stockage avant leur utilisation. Cette dernière représente l'un des éléments essentiels dans cette architecture. Une unité de stockage idéale est celle qui a une capacité infinie et qui restaure le même niveau d'énergie jusqu'à son utilisation. Par contre une unité de stockage à une capacité limitée souffre de fuites même à charge nulle. La figure 3.3(b) montre la puissance de sortie de la source d'énergie et la puissance consommée par le nœud capteur avec le temps. Cette architecture peut satisfaire aux conditions de fonctionnement neutre en énergie même si $P_S(t) \neq P_C(t) \forall t$. Avec B_0 comme énergie de stockage résiduelle initiale et avec un stockage idéal, l'inégalité suivante doit être satisfaite à tout moment pour un fonctionnement neutre en énergie sur une durée donnée T .

$$B_0 + \int_0^T [P_S(t) - P_C(t)] dt \geq 0 \quad (3.1)$$

Dans le cas d'une unité de stockage non idéale, l'inégalité inclut la puissance de fuite (P_{leak}) et la capacité de stockage finie B , et peut indiquer ce qui suit,

$$B \geq B_0 + \int_0^T [\eta P_S(t) - P_C(t) - P_{leak}(t)] dt \geq 0 \quad (3.2)$$

Avec : B : Capacité du buffer, et η : Efficacité du chargement.

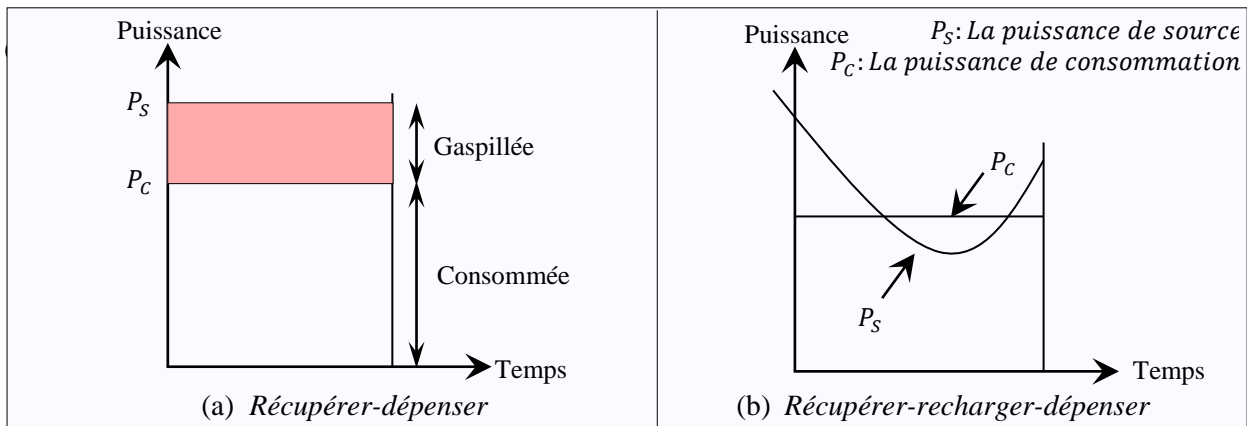


Figure 3.3: Exemples de tendances de consommation d'énergie [15]

3.7.5 Les caractéristiques de l'énergie

Il est important de noter que le *RCSF* présente des caractéristiques énergétiques très différentes de celles du *RCSF-RE*. Le niveau d'énergie dans le réseau alimenté par batterie diminue avec le temps et reste opérationnel jusqu'à ce que le niveau d'énergie soit nul. Au contraire, les nœuds appartenant à *RCSF-RE* consomment généralement plus d'énergie (dans leurs opérations courantes) qu'ils ne peuvent récolter à certaines périodes. Par conséquent, un certain niveau d'accumulation d'énergie est recommandé lors de l'utilisation du stockage avant de commencer avec le fonctionnement normal de *RCSF-RE*. Ce comportement caractéristique offre une quantité d'énergie (théoriquement) illimitée à *RCSF-RE*, ce qui les rend adaptés à de nombreuses applications énergivores [106] [107] requérant une durée de vie prolongée de la batterie. Il est inévitable que les concepteurs intègrent cette dynamique aux protocoles *MAC* spéciaux pour *RCSF-RE*.

3.7.6 Les profils de charge variables

Le protocole *MAC* joue son rôle dans la réalisation d'un contrôle optimal, juste et rapide, obtenu par la coordination des nœuds de capteurs, ce qui oblige les nœuds à rester éveillés au maximum. Comme indiqué ci-dessus, le temps de charge de tous les nœuds varie en fonction de l'environnement, du temps et du type de la récolteuse. Les nœuds s'endorment pendant qu'ils chargent suffisamment de batterie pour accomplir leurs opérations en cours. Cela impose une autre contrainte à la conception *MAC* traditionnelle qui influence directement les métriques de performance. Par conséquent, ce nouvel ensemble de considérations de conception *MAC* pour *RCSF-RE* les distingue du protocole *RCSF* standard et les protocoles *MAC* spéciaux pour *RCSF-RE* (décrits dans la section suivante) sont basés sur les nouvelles contraintes de conception imposées par les architectures de récupération d'énergie.

3.8 Les protocoles *MAC* pour les *RCSF* à récupération d'énergie

Les *RCSF* ont été utilisés dans de nombreuses applications dans des différents domaines. Le principal obstacle au développement rapide de ces réseaux est la limitation de l'énergie qui les empêche de répondre à des exigences d'application spécifiques. Récemment, les progrès de la technique de récupération d'énergie ont permis de restituer l'énergie des capteurs via des sources externes.

Les *RCSF-RE* sont passés d'un concept visionnaire à la réalité. Cependant, ce concept en est encore à ses balbutiements et nécessite une recherche approfondie pour répondre aux besoins. Pour les progrès futurs des *RCSF-RE*, la sous couche *MAC* (*Medium Access Control*) a incontestablement un rôle décisif à jouer. Un protocole *MAC* bien conçu peut gérer l'accès au canal de telle sorte que l'énergie récoltée soit utilisée efficacement et que les performances soient maximisées. Nous présentons dans cette section les caractéristiques d'un protocole *MAC* correctement conçu pour les *RCSF-RE* et fournissons un aperçu des protocoles *MAC* qui ont été explicitement conçus pour les

RCSF-RE. À la fin nous présentons une analyse théorique où on cite les avantages et les inconvénients de chaque protocole dans le but de donner une comparaison théorique entre ces protocoles *MAC* basés sur la récupération d'énergie.

3.8.1 Les protocoles *MAC* d'interrogation probabiliste pour les *RCSF-RE*

- ***PP-MAC (probabilistic polling MAC)***. Dans [108], l'interrogation probabiliste *MAC (PP-MAC)* pour les *RCSF* à un seul saut a été proposée pour améliorer les performances des canaux en termes de débit du réseau, d'équité et de temps inter-arrivé. Ce protocole prend en compte le taux de récupération d'énergie, la densité de nœuds et les collisions de paquets. Dans *PP-MAC*, le récepteur transmet périodiquement un paquet d'interrogation incluant une probabilité de contention, P_{con} . À la réception de ce paquet, chaque nœud de capteur engendre un nombre aléatoire y compris entre 0 et 1. Par la suite, le capteur transmettra son paquet de données si $y < P_{con}$; Sinon, il restera soit dans l'état de réception pour recevoir le paquet d'interrogation suivant, soit dans l'état de récolte d'énergie si son énergie est inférieure à l'exigence de transmission d'un paquet de données. Dans ce protocole, les auteurs appliquent une approche *AIMD (Additive Increase Multiplicative Decrease)* pour adapter la probabilité de contention au nombre de nœuds actifs. Si le récepteur ne reçoit aucun paquet de données en réponse à son interrogation, il augmente la probabilité de contention pour le prochain cycle d'interrogation; sinon, il diminue la probabilité de contention en cas de collision de paquets de données. Le *PP-MAC* atteignait de bons résultats tels que le débit et l'équité, en plus de fournir la flexibilité et l'évolutivité pour soutenir un grand nombre de nœuds dans un *RCSF-RE* dense. Néanmoins, ce protocole ne supporte pas les réseaux multi-saut.

- ***EH-MAC (Energy Harvesting MAC)***. Afin de prolonger le *PP-MAC* pour des scénarios multi-sauts dans les *RCSF-RE*, dans [109], les auteurs ont proposé un protocole de contrôle d'accès au médium (*EH-MAC*). Contrairement à la plupart des protocoles *MAC* existants qui sont conçus pour prolonger la durée de vie du réseau, *EH-MAC* est conçu pour atteindre un débit élevé étant donné la quantité variable d'énergie ambiante pouvant être récoltée à différents endroits et à différents moments de l'environnement. *EH-MAC* est basé sur une interrogation asynchrone, initiée par le récepteur; cependant, contrairement aux mécanismes de retrait aléatoires traditionnels, il utilise l'interrogation probabiliste pour réduire les collisions de paquets de données.

EH-MAC ajuste également dynamiquement le nombre de paquets d'interrogation pour minimiser les interférences. Chaque nœud récepteur diffuse périodiquement des paquets d'interrogation pour demander des données. Tous les nœuds d'interrogation doivent attendre un temps aléatoire compris entre 0 et t_{max} et envoyer des paquets d'interrogation uniquement si le canal est libre pendant cette période. La probabilité de contention P_{con} , est incluse dans le paquet d'interrogation qui détermine la probabilité qu'un nœud soit autorisé à émettre. Outre l'*AIMD*, les auteurs présentent *ENAN*

(Estimation du nombre de voisins actifs) pour l'ajustement de la probabilité de contention. Cette méthode est basée sur l'estimation des voisins actifs de chaque nœud récepteur. Si le nombre estimé des voisins actifs est n_{active} , la probabilité de contention sera définie sur $1/n_{active}$. En cas de collision, le récepteur augmente de n_{active} de 1 et ajuste la probabilité en conséquence. De même, lorsqu'aucun des émetteurs ne transmet leurs paquets de données, le récepteur suppose que le nombre estimé est supérieur au nombre réel de voisins actifs. Ainsi, le nombre de nœuds est réduit de 1 et la probabilité de contention augmente. Un exemple illustrant le processus de réception des paquets de données et de mise à jour des n_{active} est présenté à la figure 3.4; les voisins des nœuds sont indiqués entre parenthèses.

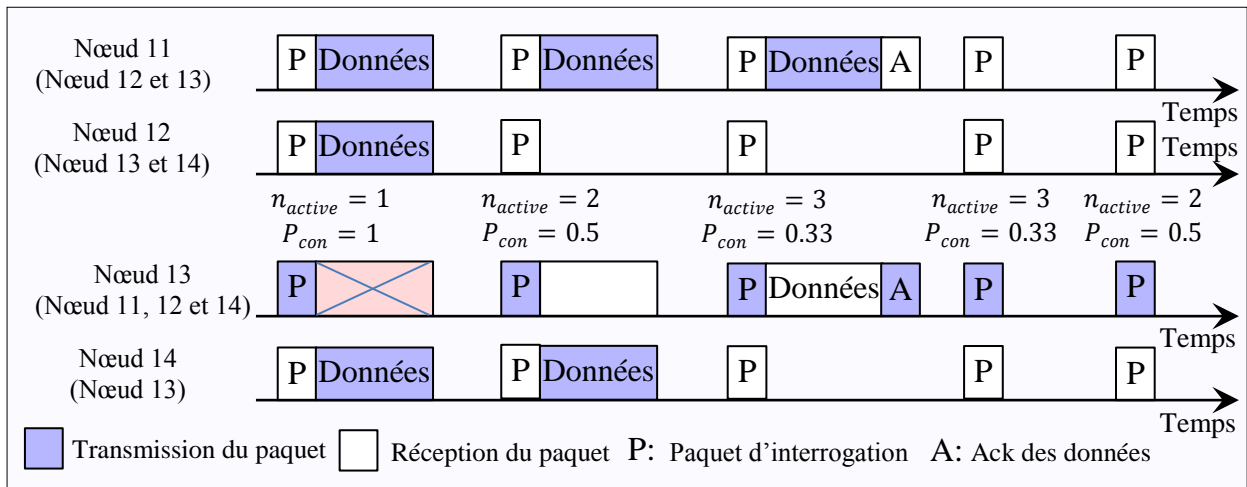


Figure 3.4: Exemple de processus d'EH-MAC [109]

▪ **MTPP (multi-tier probabilistic polling scheme).** Dans [110], Chisato et Winston ont proposé un système d'interrogation probabiliste à plusieurs niveaux (MTPP) pour la fourniture de données dans les réseaux multi-sauts qui reposent uniquement sur RE comme source d'énergie, qui est une nouvelle approche pour prolonger PP-MAC à multi-saut qui utilise un modèle hiérarchique basé sur les niveaux. Dans cette proposition, chaque niveau consiste en un ensemble de nœuds dépendant de la distance de sink.

De manière similaire à PP-MAC, dans un RCSF avec trois niveaux par exemple comme il est indiqué dans la figure 3.5, le récepteur (le sink) diffuse périodiquement aux nœuds de niveau 1 un paquet d'interrogation contenant la valeur de la probabilité de contention. Lors de la réception de ce paquet, un nœud du niveau 1 est interrogé pour diffuser la valeur probabiliste identique au niveau 2; il attend ensuite de recevoir un paquet de données et par la suite ce même processus est exécuté avec un nœud du niveau 2 au niveau supérieur afin de recevoir un paquet de données qui va être transmis à la fin vers le sink. En revanche, pour les propositions existantes qui reposent sur des simulations, les auteurs ont mis en œuvre le schéma sur du matériel commercial disponible sur le marché avec uniquement une échelle de deux niveaux et l'ont validé expérimentalement pour démontrer la viabilité de la proposition.

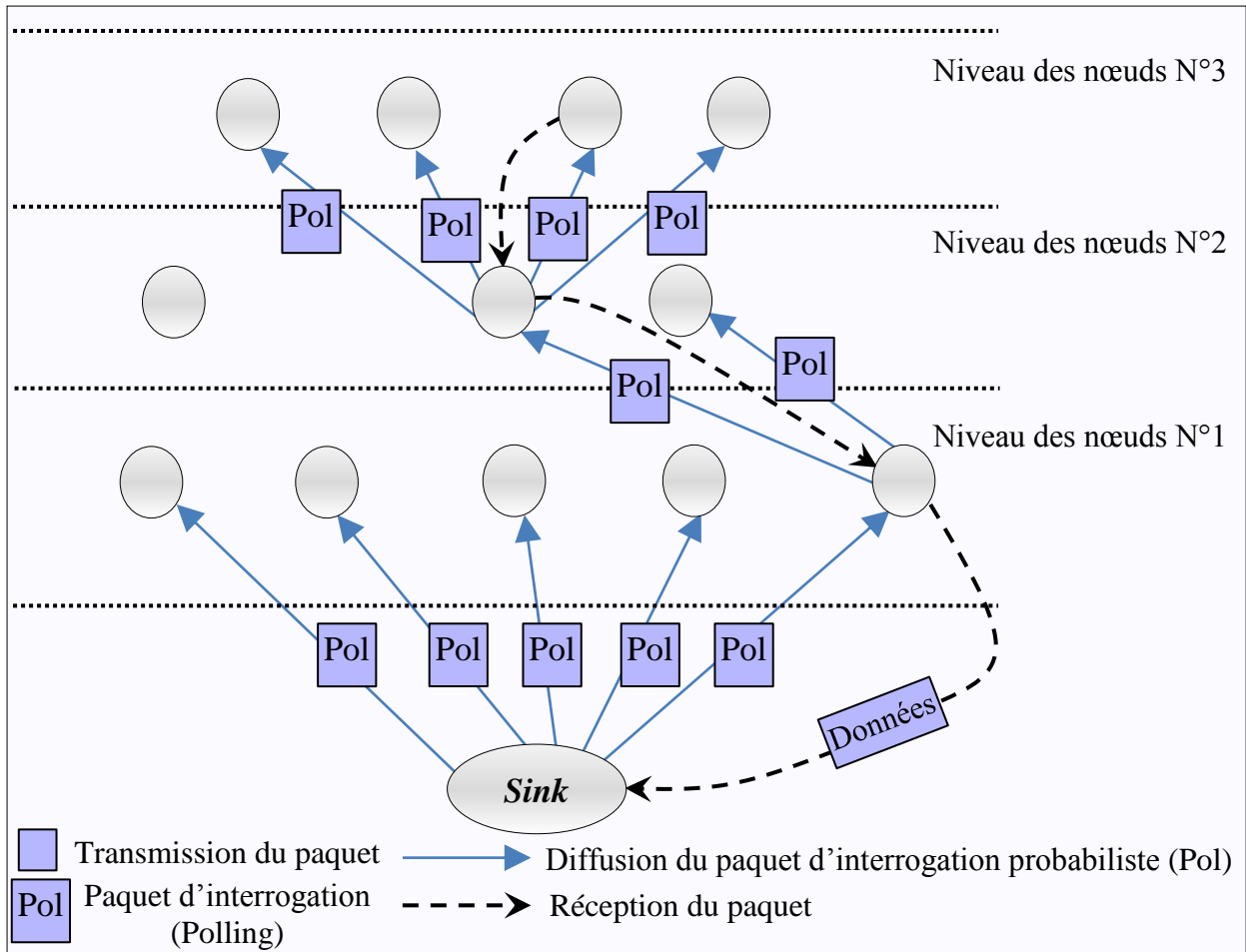


Figure 3.5: Exemple de processus de *MTPP* avec un *RCSF* à 3 niveaux [110]

3.8.2 Les protocoles MAC d'interrogation non probabiliste pour les *RCSF-RE*

▪ ***OD-MAC (on demand MAC protocol)***. Les auteurs ont développé dans [111], un protocole *MAC* à la demande (*OD-MAC*) pour les *RCSF-RE*, qui utilise une communication initiée par le récepteur et qui est basé sur la notion de *duty-cycle*. Le protocole en question est capable de prendre en charge des *duty-cycles* individuels pour des nœuds ayant différents profils d'énergie. Ainsi, chaque nœud est capable d'augmenter sa consommation d'énergie, ainsi que ses performances, tant que l'énergie consommée est égale à la quantité d'énergie récoltée (e.g., conserver le nœud en état *ENO-MAX*). Dans *OD-MAC*, chaque nœud récepteur diffuse périodiquement une balise de beacon vers les émetteurs, affirmant sa décision de recueillir des paquets de données. Après la réception de cette balise par ces émetteurs, la transmission des paquets de données aura lieu. Un nœud qui trouve un excès d'énergie récupéré peut réduire l'intervalle de transmission de balise (le temps entre deux transmissions de balise consécutives) ou l'intervalle de détection (le temps entre deux opérations de détection consécutives) en fonction des besoins de l'application. En cas de pénurie d'énergie, le nœud peut augmenter l'un ou l'autre des intervalles. Lors du transfert opportuniste, chaque nœud conserve une liste de nœuds de relais potentiels et envoie

son paquet dès qu'il reçoit une balise de l'un des membres de la liste. Cela élimine le besoin d'attendre un récepteur particulier. Dans [112], un nouveau mode de fonctionnement, appelé mode de liaison, est défini pour les scénarios dans lesquels l'énergie d'entrée est extrêmement faible. Dans des conditions de faible énergie, un nœud émetteur abandonne le transfert opportuniste et passe en mode liaison dans lequel il sélectionne un récepteur particulier et se lie à son cycle de fonctionnement. La figure 3.6 illustre le fonctionnement du protocole *OD-MAC*.

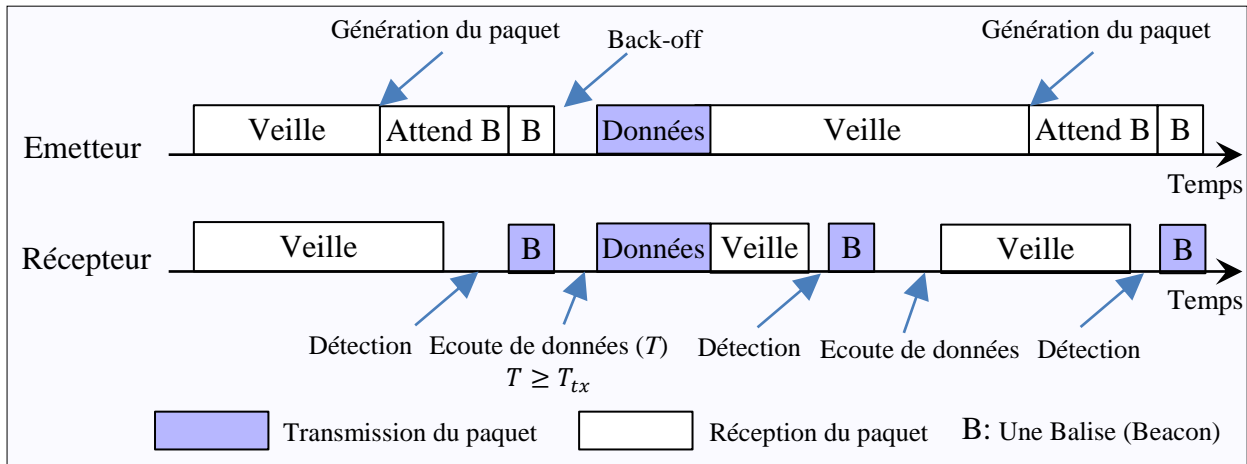


Figure 3.6: Illustration du fonctionnement du protocole *OD-MAC* [111]

▪ ***ERI-MAC (Energy-Harvested Receiver-Initiated MAC Protocol)*.** Les auteurs de [113] ont proposé *ERI-MAC*, qui est un nouveau protocole *MAC* initié par un récepteur pour les réseaux de capteurs à récupération d'énergie et qui est basé sur l'utilisation du protocole *CSMA/CA* pour l'accès au canal. Afin d'atteindre de bonnes performances en latence et en efficacité énergétique, *ERI-MAC* exploite l'avantage de la concaténation de paquets initiée par le récepteur. En outre, *ERI-MAC* utilise une technique de mise en file d'attente pour ajuster le comportement opérationnel d'un nœud de capteur en fonction du taux de récupération d'énergie de l'environnement voisin, obtenant ainsi l'état *ENO*, où chaque paquet en attente est retardé pendant une durée sûre pour s'assurer que l'énergie consommée est inférieure à l'énergie récoltée. Le rapport entre l'énergie consommée et l'énergie récoltée est vérifié après la durée de sécurité. Par cette comparaison, un nœud détermine s'il dépasse le niveau d'énergie suffisant.

Le schéma de communication de base utilisé par *ERI-MAC* est similaire à celle de *PP-MAC* et *OD-MAC*. Lorsqu'un nœud se réveille et qu'aucun paquet n'est programmé pour être transmis, le nœud diffuse un paquet de balise contenant son propre identifiant. Après qu'un expéditeur a reçu une balise attendue, une transmission de paquet commence avec le paquet en tête de la première file d'attente en cours de transmission. Comme dans *EH-MAC*, un paquet d'accusé de réception est envoyé pour confirmer que le paquet a bien été reçu, qui est également utilisé comme nouvelle balise. La figure 3.7 montre un exemple d'opération de communication entre un

expéditeur et un destinataire. Le protocole *ERI-MAC* est évalué sur une topologie de grille de 49 nœuds avec une distance fixe entre deux voisins.

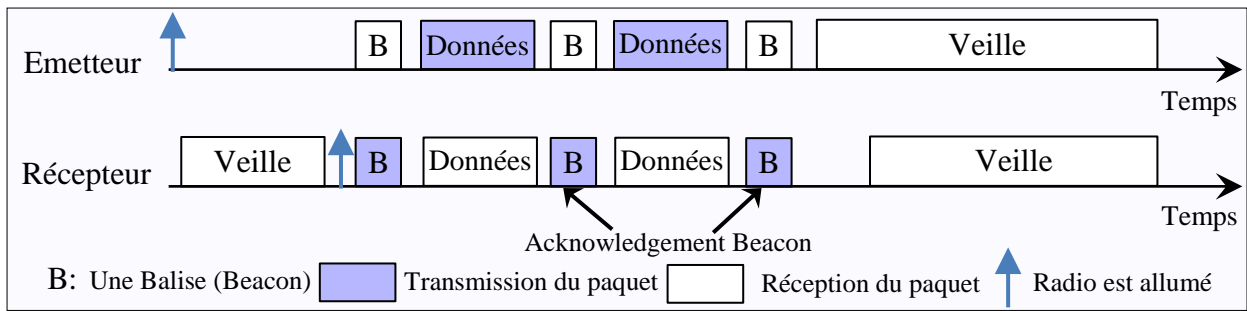


Figure 3.7: Illustration du fonctionnement du protocole *ERI-MAC* [113]

▪ ***QAEE-MAC (QoS Aware Energy-Efficient MAC Protocol)***. Dans [114], les auteurs ont développé un protocole *MAC* de *QoS* pour les *RCSF-RE* appelé protocole *QAEE-MAC*, qui est un protocole *MAC* asynchrone initié par le récepteur, qui permet la transmission plus rapide des paquets de données urgents que des paquets normaux [115]. Pour améliorer la durabilité du réseau avec efficacité, *QAEE-MAC* tire parti d'un mécanisme de priorité des données pour sélectionner le nœud émetteur. Un nœud récepteur se réveille périodiquement, diffuse une trame de beacon contenant les adresses des émetteurs sélectionnés après avoir d'abord reçu les trames de beacon d'expéditeur où il sélectionne le nœud émetteur en fonction de la priorité des données des expéditeurs. L'expéditeur ayant la priorité la plus élevée commence à transmettre son paquet alors que d'autres vont dans un état de veille.

Comme dans *ERI-MAC*, une balise d'accusé de réception transmis par le récepteur indiquant la réception réussie d'un paquet de données précédemment reçu. Un exemple de processus de transmission de données prioritaire est présenté à la Fig. 3.8.

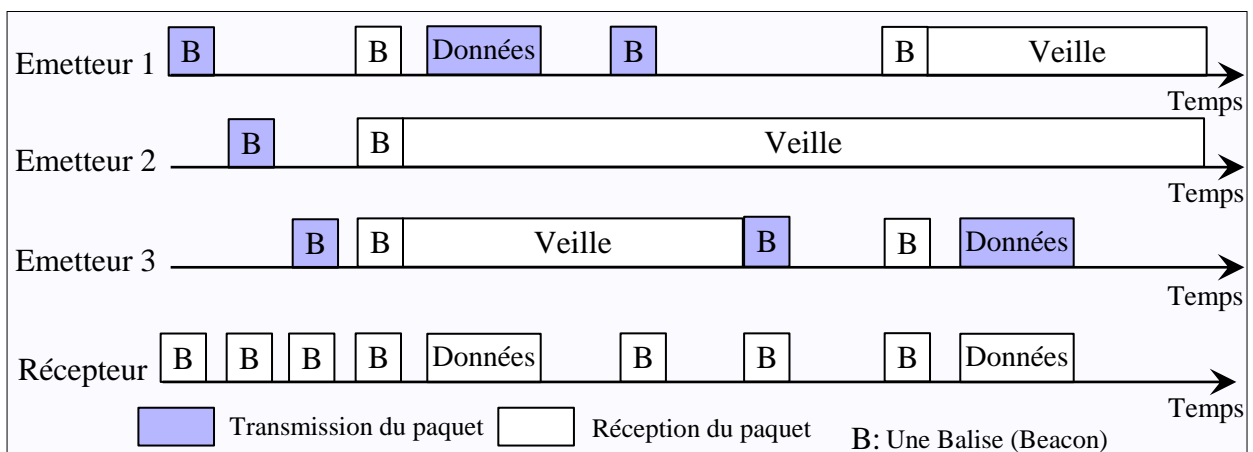


Figure 3.8: Exemple de processus de transmission de données prioritaire dans *QAEE-MAC* [114]

De plus, *QAEE-MAC* ajuste également sa période de réveil en fonction de l'état énergétique. Par conséquent, il minimise sa consommation d'énergie. L'évaluation des performances du protocole *QAEE-MAC* est limitée à un seul récepteur et à un petit

nombre de nœuds émetteurs d'une manière unique. Ce protocole améliore à la fois le débit et la livraison de paquets de bout en bout.

▪ **P-MAC (Pulsed-MAC Protocol).** Layerle et Kwasinski [116] ont développé un protocole MAC pour les RCSF-RE appelé P-MAC qui est présenté pour administrer efficacement les communications sans fil dans les réseaux corporels (*BAN: Body Area Networks*). Le P-MAC prolonge en particulier la durée de vie du réseau en améliorant les nœuds de capteurs grâce à l'intégration d'un circuit de pompage de charge qui récupère l'énergie à partir du signal pulsé utilisé pour réveiller les nœuds. La principale stratégie de P-MAC consiste à concentrer les processus de communication dans la station de base. Ce faisant, nous permettons aux nœuds de capteurs de prolonger leur durée de vie en limitant l'utilisation de l'énergie précieuse uniquement lorsque cela est absolument nécessaire.

Spécifiquement, avec le P-MAC, un signal d'impulsion est transmis par la station de base aux nœuds du capteur. Chaque nœud est conçu avec un circuit de pompage de charge capable de capter l'énergie dans l'impulsion. Par conséquent, la détection des impulsions ne consomme aucune des précieuses ressources du nœud. Le choix de concentrer les processus de communication dans la station de base est jugé acceptable, car une station de base type pourrait être conçue comme le point unique nécessitant un rechargement facile. Dans [117], une impulsion de 10 mW a été détectée avec succès à environ 3 m sans aucun circuit amplificateur, ce qui est suffisant pour les BAN. Transmission à 10 mW a été considéré comme sûre pour le patient BAN portant, comme ce pouvoir est typique pour les autres appareils portables sans fil (par exemple, les téléphones cellulaires) et dans les limites de la FCC (*Federal Communications Commission*) [10].

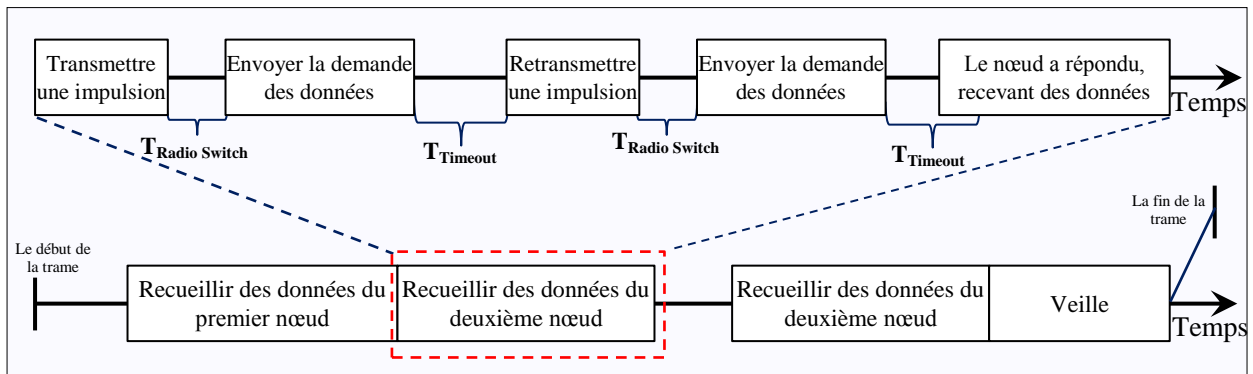


Figure 3.9: Le fonctionnement de la station de base pendant un temps trame [116]

Dans P-MAC, un codage à intervalles d'impulsions *PIE (Pulse Interval Encoding)* est utilisé pour permettre l'adressage de nœuds. L'adressage est essentiel pour économiser autant d'énergie que possible car il permet à la station de base de n'avoir qu'un seul nœud spécifique avec sa radio allumée à un moment donné. Une fois que la station de base a communiqué avec chaque nœud du réseau, elle se met en veille jusqu'à ce qu'une nouvelle trame démarre. Une chronologie décrivant le

fonctionnement de la station de base pendant un temps de trame est illustrée dans la figure 3.9.

▪ **EL-MAC (Energy Level based MAC Protocol).** La radio cognitive est une technologie qui permet aux émetteurs-récepteurs de détecter les canaux vacants temporairement et de les exploiter de manière sophistiquée afin d'optimiser l'utilisation du spectre, sans interférer avec la transmission des autres utilisateurs. Le fonctionnement cognitif peut être effectué dans des bandes sous licence utilisées pour découvrir et accéder de manière opportuniste à des trous de spectre. Les utilisateurs autorisés disposant d'une licence d'utilisation du spectre sont appelés utilisateurs primaires et ceux qui utilisent la radio cognitive pour trouver une opportunité d'accès sont appelés utilisateurs secondaires. Ces derniers doivent exécuter l'action cognitive avec soin pour éviter d'interférer avec les transmissions en utilisateurs primaires.

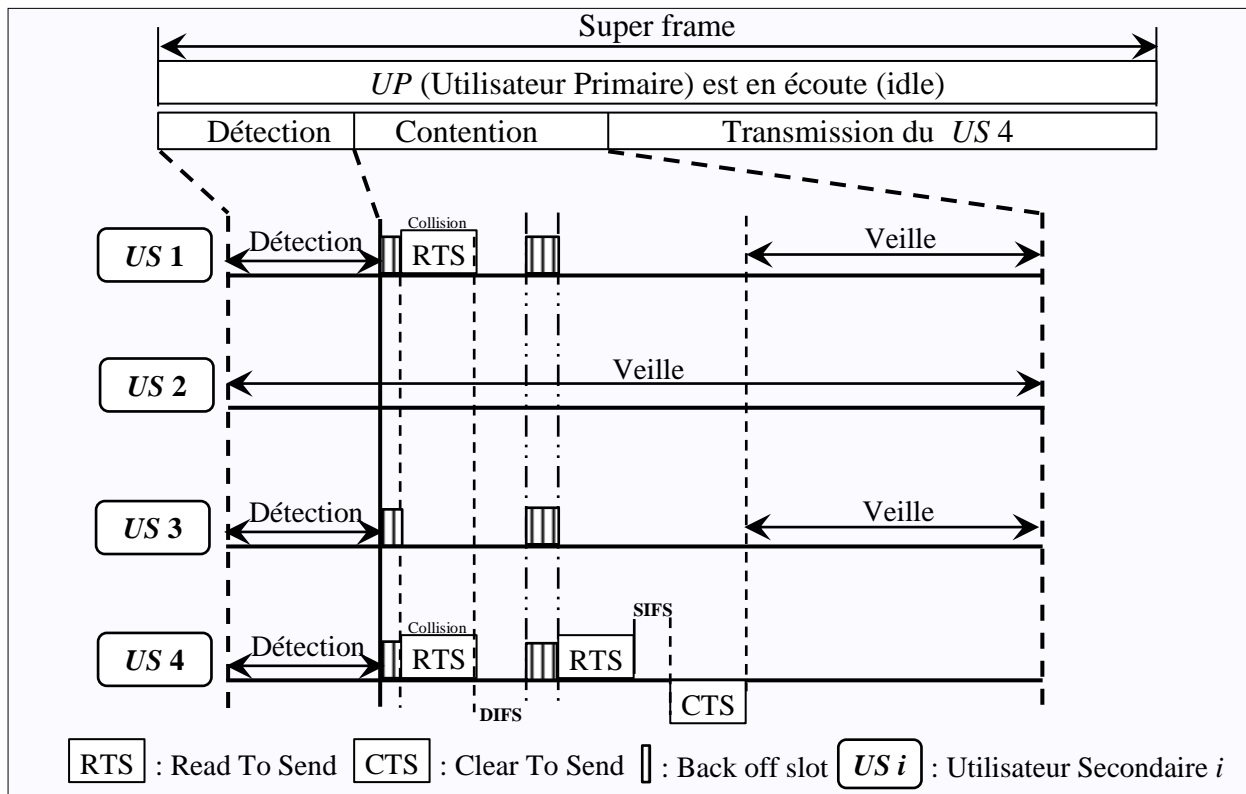


Figure 3.10: Un exemple du protocole EL-MAC [118]

Les auteurs de [118] proposent un protocole MAC basé sur le niveau d'énergie appelé EL-MAC. EL-MAC est introduit pour les utilisateurs de la récupération d'énergie dans les réseaux radio cognitive. Il utilise deux types d'utilisateurs dans un environnement à canal unique; les utilisateurs primaires (UP) à haute énergie et les utilisateurs secondaires (US) à basse énergie, tandis que la communication est réalisée sur une base de super trame. Ce dernier est divisé en trois périodes: la période de détection, la période de contention et la période de transmission. Principalement, dans EL-MAC, une priorité élevée est accordée aux US, de sorte qu'il peut accéder rapidement au canal tandis que les UP peuvent transmettre des données après que les

US sont passés en mode veille après la transmission. Les *US* peuvent vérifier si le canal est utilisé par les *UP* ou non selon la probabilité d'accès au canal en fonction de son niveau d'énergie. Si le canal est libre, les *US* peuvent passer à la période de contention où le protocole *CSMA/CA* (*Carrier Sense Multiple Access with Collision Avoidance*) est utilisé et par la suite, la transmission est effectuée; sinon, les *SU* se mettent immédiatement en veille pour ne pas déranger les communications des *UP*. Par conséquent, *EL-MAC* augmente l'efficacité énergétique. Un exemple illustrant le processus du protocole *EL-MAC* présenté à la figure 3.10.

▪ ***LEB-MAC (Load and Energy Balancing MAC Protocol)***. Dans [119], les auteurs ont proposé un protocole *MAC* d'équilibrage de charge et d'énergie pour les réseaux de capteurs sans fil alimentés par récupération d'énergie (*LEB-MAC*), qui est un autre protocole *MAC* initié par le récepteur pour les *RCSF-RE*. Dans *LEB-MAC*, la performance des *RCSF* alimentés par récupération d'énergie est améliorée sous trois aspects: le relais, l'ordonnancement et le contrôle d'accès au support. Les balises transmises par les nœuds récepteurs transmettent les informations sur le programme de réveil suivant qui permet au nœud émetteur de réduire son duty-cycle et de réguler les intervalles de réveil et de sommeil en fonction de ses niveaux d'énergie. Cependant, si le nœud expéditeur n'a aucune information sur la planification du destinataire, il peut être forcé d'attendre longtemps pour recevoir une balise de ce nœud récepteur où le temps d'attente maximal (SL_{max} (durée maximale de veille d'un récepteur)) de l'expéditeur est lié à l'application.

Le duty-cycle d'un nœud récepteur est ajusté en fonction de son niveau d'énergie. Par conséquent, un nœud qui fait face à une déficience énergétique dort plus longtemps pour pouvoir récolter suffisamment d'énergie avant l'heure de réveil suivante. Les auteurs exploitent le concept d'une logique floue pour définir les intervalles de sommeil des nœuds en fonction des niveaux d'énergie. Les nœuds avec une énergie résiduelle plus élevée se réveillent plus fréquemment, de sorte qu'un nœud émetteur ayant plus d'une option pour le transfert de paquets peut choisir arbitrairement l'un des récepteurs potentiels actifs pour le relais de paquets. Ceci est quelque peu similaire au schéma de transfert opportuniste d'*OD-MAC* car les nœuds n'ont pas à attendre une balise spécifique. De plus, des vides dans le réseau peuvent apparaître lorsque les nœuds ont besoin de temps pour se recharger; forçant ainsi les paquets de données à être transmis via d'autres chemins. Cette stratégie de relais vise à prévenir les trous en équilibrant la charge sur le réseau en fonction des caractéristiques de récupération d'énergie des nœuds.

Dans l'exemple de la figure 3.11, la liste des récepteurs du nœud *E* est nulle s'il communique pour la première fois après un démarrage à froid; un nœud dans *RCSF-RE* commence à partir d'un démarrage à froid lorsqu'il a déjà dépensé toute son énergie stockée et perdu toutes les informations d'état. Après avoir reçu des balises du récepteur 2, 1 et 3, la liste des récepteurs du nœud *E* devient (2, 1, 3). Avec les

informations acquises sur les plannings des nœuds voisins, le nœud E sait maintenant quand se réveiller pour attendre la prochaine balise. Pour les transmissions de paquets suivantes, le nœud E se réveille juste avant le temps de transmission attendu d'une balise, comme indiqué sur la Fig. 3.11. Cependant, bien que rares, plusieurs récepteurs peuvent envoyer simultanément des balises. Une procédure de résolution de collision de balise (ainsi que de balise manquée) est donc invoquée pour résoudre ce problème.

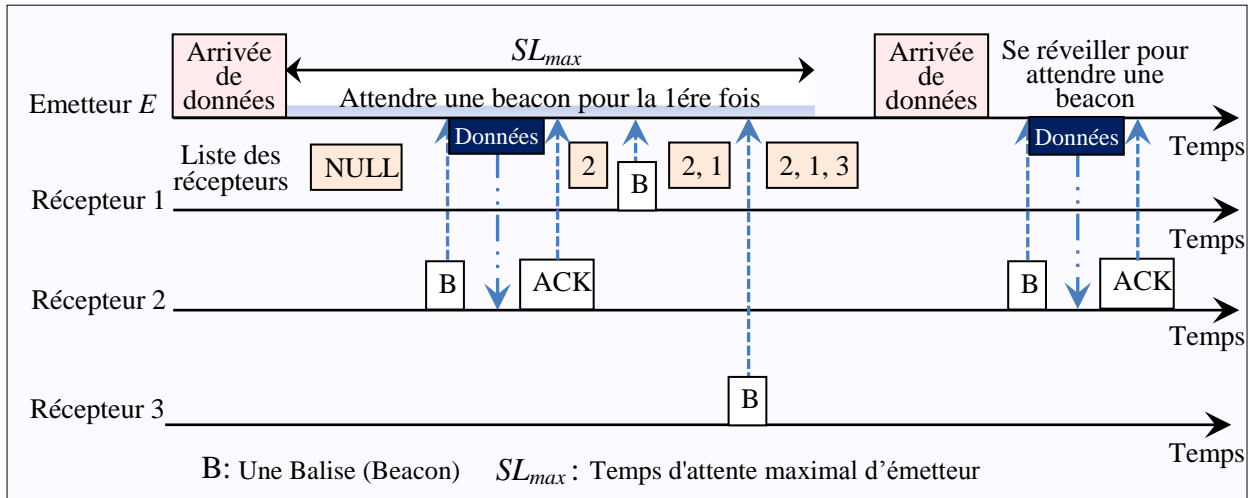


Figure 3.11: Illustration du fonctionnement du protocole LEB-MAC [119]

▪ **DeepSleep (IEEE 802.11 Enhancement for EH M2M communications).** La communication Machine à machine ($M2M$) est devenue un nouveau paradigme de réseau important et permet l'échange continu d'informations entre les appareils autonomes, comme la conception du protocole MAC pour $M2M$ est très similaire à celle du $RCSF$, la majorité des schémas proposés pour la communication $M2M$ sont inspirés par les protocoles MAC des $RCSF$.

Dans [120], les auteurs ont proposé le mécanisme *DeepSleep*, un schéma d'amélioration MAC sur le mode d'économie d'énergie (*Power Saving Mode (PSM)*) *IEEE 802.11*, pour prendre en charge un réseau de communication $M2M$ sans fil avec de grandes topologies composées de dispositifs autonomes de récupération d'énergie. *DeepSleep* tire parti des propriétés apportées par la technologie de récupération d'énergie pour recharger l'énergie des dispositifs. Dans *DeepSleep*, le temps est divisé en intervalles de balise (BI) et chaque BI se compose d'une fenêtre $ATIM$ (*Ad hoc Traffic Indication Map*) et d'un intervalle de transmission de données. Lorsqu'un dispositif a des données destinées à un autre dispositif, il envoie d'abord un message de demande $ATIM$ et attend un accusé de réception $ATIM$. Les paires, qui ont échangé avec succès une requête $ATIM$ et un accusé de réception, restent éveillées pour le reste de la BI et d'autres dispositifs passent en mode veille pour économiser de l'énergie. Ce mécanisme d'économie d'énergie de la norme *IEEE 802.11* est efficace tant que le nombre de périphériques est limité. Toutefois, dans les réseaux à grande échelle, le mécanisme est soumis à de graves conflits qui se traduisent par un gaspillage d'énergie dû à l'écoute, à l'écoute *inactive* et aux collisions. *DeepSleep* est conçu pour résoudre

ce problème grâce à ses mécanismes *Energy-Aware Deep Sleeping* (mode veille avancée) et *Controlled Access* (Accès contrôlé).

En *Energy-Aware Deep Sleeping*, un *Deep Sleep* (Sommeil profond) est activé lorsque le niveau d'énergie d'un dispositif tombe en dessous du seuil $E_{DeepSleep}$. Le dispositif dort profondément pour les intervalles de balise n_{BP} (*BP* : Beacon Period), tout en récoltant l'énergie des sources ambiantes. Pour améliorer encore les performances du réseau dans les réseaux *M2M* les plus encombrés, *DeepSleep* avec accès contrôlé est proposé. Dans Access contrôlé, lorsqu'un dispositif se réveille et trouve des paquets dans son tampon, il doit d'abord générer un nombre aléatoire compris entre 0 et 1 et le comparer à la valeur prédéfinie de P_{Sleep} . Un événement *DeepSleep* est déclenché si le nombre généré est inférieur à P_{Sleep} . Dans les deux mécanismes, les périphériques qui se réveillent après avoir dormi profondément se verront attribuer des fenêtres de contention plus petites afin de garantir leur priorité d'accès au canal.

▪ ***DSR (Duty-cycle Scheduling based on Residual energy) et DSP (Duty-cycle Scheduling based on Prospective increase in residual energy)***. Des schémas différents et plus pratiques ont été proposés par Yoo et al. [121]. Planification duty-cycle basée sur l'énergie résiduelle (*DSR*), et une approche plus complexe: Planification du duty-cycle basée sur l'augmentation potentielle de l'énergie résiduelle (*DSP*). *DSR* détermine l'allocation d'énergie de nœud exprimée en termes d'intervalle de réveil uniquement en fonction du niveau de batterie ou de la charge résiduelle. L'intervalle de réveil est indiqué par I_{dc}^i et il est calculé par:

$$I_{dc}^i = I_{dc}^{max} - \left(I_{dc}^i \cdot \left(\frac{E_r^i - E_{th}}{E_{max} - E_{th}} \right) \right) \quad (3.3)$$

Où E_r^i est le niveau d'énergie actuel, E_{th} est le niveau de batterie minimal utilisé comme seuil, I_{dc}^{max} et E_{max} représentent l'intervalle de réveil maximal et le niveau de batterie maximal respectivement.

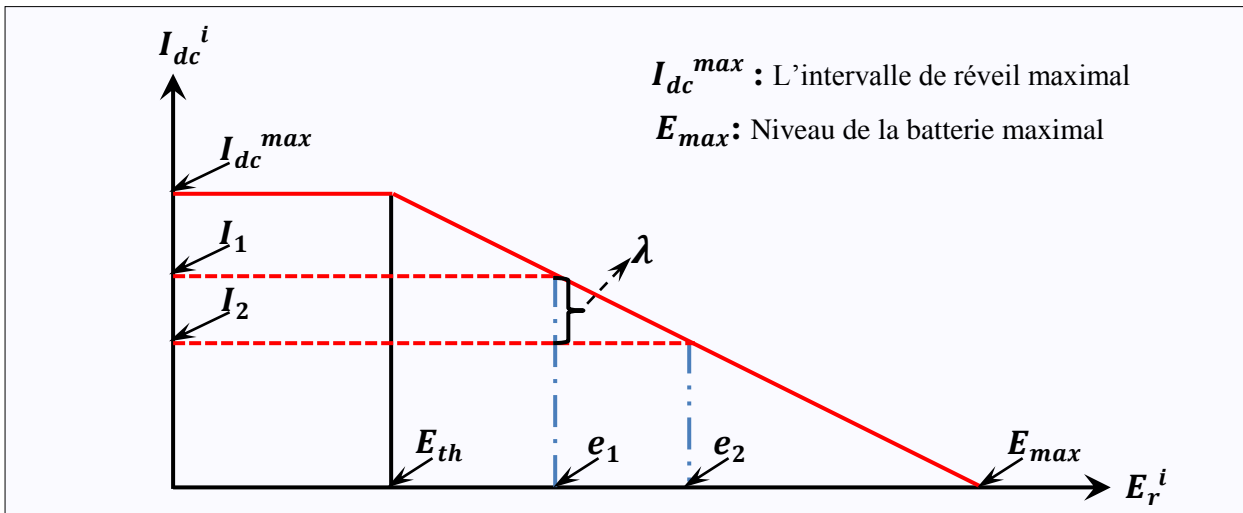


Figure 3.12: Graphe de décision du *DSR* et *DSP* [121]

La figure 3.12 montre le graphe de décision de *DSR*. Sur l'axe des abscisses nous avons le niveau d'énergie et sur l'axe des y nous avons l'intervalle de réveil. Nous pouvons remarquer qu'avec cette procédure, nous avons une relation entre le niveau de la batterie et l'intervalle de réveil. Dès que le niveau de la batterie augmente, l'intervalle de réveil est réduit (plus de consommation d'énergie). Sinon, si le niveau de la batterie diminue, l'algorithme augmente l'intervalle de réveil pour réduire la consommation d'énergie.

Le *DSP* est une approche plus agressive similaire au *DSR*: dès qu'un nœud sait qu'il y a une augmentation dans le niveau de leur batterie, il peut réduire le temps de réveil compatible avec le nouveau niveau de batterie prévu. En d'autres termes, si un nœud sait que le niveau de batterie augmentera dans l'emplacement suivant, il utilisera un intervalle de réveil plus court. Pour en savoir plus sur l'augmentation de la perspective du niveau de la batterie, le nœud peut vérifier le niveau de la batterie ou vérifier la variation de la source de récolte. Pour éviter l'épuisement exagéré de la batterie, dès que la batterie diminue, le *DSP* revient au comportement du *DSR* pour être moins agressif et réduire la consommation d'énergie.

Par conséquent, les deux schémas *DSR* et *DSP* sont introduits afin de réduire le temps de latence du sommeil (c-à-d. le temps d'attente requis par un émetteur avant de transmettre le paquet) de nœuds de capteur tout en maintenant un équilibre de consommation d'énergie entre les nœuds de capteurs.

3.9 Analyse théorique de différents schémas MAC pour RCSF-RE

La récupération d'énergie est une technologie prometteuse utilisée pour améliorer la performance et la durée de vie des réseaux de capteurs sans fil, car elle évite de devoir remplacer les sources d'énergie et atténue également la limitation de l'énergie dans le réseau. De ce point de vue, différents schémas MAC asynchrones basés sur un paradigme de balisage ont été proposés dans le but d'exploiter l'énergie ambiante, qui peuvent être classée en deux catégories selon le mécanisme de contention sur le canal utilisé par les nœuds capteurs dans le réseau: le premier utilise le mécanisme d'interrogation probabiliste tels que (*PPMAC*) et (*EH-MAC, MTPP*) pour le réseau à un-saut et à multi-saut respectivement tandis que le second utilise le protocole *CSMA/CA* tels que (*QAEE-MAC, EL-MAC* et *DeepSleep*) et (*ODMAC, ERI-MAC, LEB-MAC, DSR* et *DSP*) pour les réseaux à un et à multiples sauts respectivement. Les avantages de chaque classe sont:

1. Pour la première classe

- 1- un débit amélioré grâce au schéma d'interrogation probabiliste utilisé pour réduire les collisions de paquets dans le réseau;
- 2- une équité élevée car tous les nœuds ont la même probabilité d'accéder au canal dans le réseau;
- 3- une durée de vie élevée du réseau grâce à l'équilibrage de la consommation d'énergie qui est basé sur la quantité d'énergie récoltée et le système d'interrogation probabiliste utilisé.

2. Pour la deuxième classe

- 1- la probabilité de collision est réduite dans le réseau en forçant certains nœuds actifs à s'endormir et abandonner la contention sur le canal avec des intervalles de contention différents prédéfinis pour divers nœuds afin de déterminer les priorités de certains sur d'autres comme dans *EL-MAC* et *DeepSleep*. En conséquence, la latence est modérée et le débit est élevé mais l'équité est élevée, uniquement dans *DeepSleep*;
- 2- du fait du paramétrage du duty-cycle des nœuds en fonction de leur niveau d'énergie et de leur capacité de récupération d'énergie, la latence en (*ODMAC*, *ERIMAC*, *QAEE-MAC*, *LEB-MAC*, *DSR* et *DSP*) est faible, le débit est élevé sauf en (*ODMAC* et *LEB-MAC*) où il est modéré et l'équité est modérée. Ce dernier est assuré par la répartition de la charge entre les nœuds. En outre, dans *LEB-MAC*, les nœuds reconnaissent le temps de réveil de leurs récepteurs et les calendriers de réveil sont définis en fonction des niveaux d'énergie actuels. De plus, dans *QAEE-MAC*, un nœud récepteur choisit un nœud émetteur en fonction de la priorité des données de l'expéditeur;
- 3- *DSR* et *DSP*, permettent aux nœuds de capteurs d'ajuster leur duty-cycle dynamiquement en fonction de leur énergie résiduelle.

Cependant, les deux classes manifestent les inconvénients suivants :

1. Pour la première classe

- 1- certains paquets peuvent être soumis à un long délai avant leur transmission malgré la disponibilité de ressources réseau suffisantes; cela peut être causé par la faible probabilité de contention et la longue durée de résidence des paquets dans chaque saut; ce qui entraîne une latence élevée dans un scénario multi-sauts et une latence modérée dans des scénarios à saut unique; de plus, dans ce cas, l'énergie et la bande passante sont gaspillées sans obtenir de résultat bénéfique jusqu'à ce que la probabilité de contention appropriée soit atteinte;
- 2- tous les schémas proposés dans cette classe n'utilisent aucune approche d'approximation dans leur ajustement du duty-cycle. Les nœuds modifient leurs intervalles de duty-cycle sans prédiction de l'énergie présentée dans le futur.

2. Pour la deuxième classe

- 1- les schémas de la première sous-classe, *ODMAC*, *ERIMAC*, *QAEE-MAC*, *EL-MAC*, *DeepSleep* et *LEB-MAC* n'utilisent aucune méthodologie d'estimation dans leur ajustement du duty-cycle. Les nœuds modifient leurs intervalles de cycle de fonctionnement sans existence consciente de l'énergie disponible en futur;
- 2- dans *EL-MAC*, l'équité est faible parce que certains utilisateurs de haute énergie cessent de se disputer, s'endorment non seulement pour conserver leur énergie mais aussi pour récolter de l'énergie supplémentaire. Ces opérateurs resteront

éveillés pour les super trames successives et ils perdront leurs probabilités pour l'envoi de données à plusieurs reprises;

- 3- bien que le duty-cycle utilisé dans les schémas *DSR* et *DSP* soit dynamique, la scalabilité et l'équité sont modérées.

Schéma	RE	TC	AC	D	L	S	E	TI
PP-MAC	Récupérer-dépenser	Un-saut	Polling	Élevé	Modérée	Élevée	Élevée	Simulation
EH-MAC	Récupérer-dépenser	Multi-saut	Polling	Élevé	Élevée	Élevée	Élevée	Simulation
MTPP	Récupérer-dépenser	Multi-saut	Polling	Élevé	Élevée	Élevée	Élevée	Test réel
OD-MAC	Récupérer-recharger-dépenser	Multi-saut	CSMA/CA	Modéré	Faible	Faible	Modérée	Simulation
ERI-MAC	Récupérer-recharger-dépenser	Multi-saut	CSMA/CA	Élevé	Faible	Élevée	Modérée	Simulation
QAEE-MAC	Récupérer-recharger-dépenser	Un-saut	CSMA/CA	Élevé	Faible	Faible	Modérée	Simulation
P-MAC	Récupérer-recharger-dépenser	Un-saut	Pulser	Élevé	Faible	Modérée	Élevée	Simulation
EL-MAC	Récupérer-recharger-dépenser	Un-saut	CSMA/CA	Élevé	Modérée	Modérée	Faible	Simulation
LEB-MAC	Récupérer-recharger-dépenser	Multi-saut	CSMA/CA	Modéré	Faible	Faible	Modérée	Simulation
Deep-Sleep	Récupérer-recharger-dépenser	Un-saut	CSMA/CA	Élevé	Modérée	Élevée	Élevée	Simulation
DSR-RI et DSP-RI	Récupérer-recharger-dépenser	Multi-saut	CSMA/CA	Élevé	Faible	Modérée	Modérée	Simulation

Tableau 3.1: La comparaison des différents schémas MAC pour RCSF-RE (Concernant leur prise en charge des différents facteurs : de la stratégie de la Récupération d'Énergie (RE), du Type de Communication (TC), de l'Accès au Canal (AC), du Débit (D), du temps de Latence (L), de Scalabilité (S), de l'Équité (E) et du Type d'Implémentation (TI)).

En plus des schémas proposés dans les deux catégories examinées, il y a un autre protocole *P-MAC* qui fonctionne avec une grande équité, un débit élevé, une faible

latence et une évolutivité modérée, implémenté dans un réseau à un seul saut. Ce protocole utilise la technique de la pulsation pour que les nœuds puissent accéder au canal. Pour conclure cette section, le tableau 3.1 est une comparaison qui résume les contributions les plus importantes dans ce domaine d'un point de vue théorique. De l'analyse ci-dessus, nous ne pouvons pas conclure quelle est la meilleure solution car chacun présente des avantages et des inconvénients. Mais il convient de noter que la plupart de ces schémas proposés n'utilisent aucune technique d'estimation pour ajuster le duty-cycle des nœuds dans le réseau.

3.10 Conclusion

La récupération d'énergie est une technologie prometteuse pour les *RCSF* et suscite de plus en plus d'intérêt, car elle permet la conversion de l'énergie ambiante présente dans l'environnement en énergie électrique employable ou stockée dans des batteries afin de servir au fonctionnement des nœuds de capteurs. Bien que cette nouvelle technologie ait besoin des développements additionnels pour atteindre le niveau où elle peut être largement utilisée dans les *RCSF*. Dans ce chapitre, nous avons présenté un bref historique sur la récupération d'énergie dans *RCSF*. Compte tenu du rôle important de la sous couche *MAC* dans les *RCSF*, nous avons décrit de nombreux schémas de *MAC* proposés pour les *RCSF-RE* avec leurs caractéristiques principales et les avons classés en deux classes suivant le mécanisme de contention sur le canal utilisé par les nœuds capteur dans le réseau; les protocoles de la première classe utilisent le mécanisme d'interrogation probabiliste et les protocoles de la deuxième classe utilisent le protocole *CSMA/CA*. La conception presque de ces protocoles est inspirée de l'architecture initiée par le récepteur. Les caractéristiques et principes de fonctionnement de ces protocoles *MAC* ont été discutés. Enfin, après avoir discuté de plusieurs protocoles, nous avons présenté une comparaison de ces protocoles d'un point de vue théorique pour différentes mesures dans le tableau 3.1, ce qui montre que la plupart des protocoles sont utilisés pour les systèmes de récupération d'énergie « *Récupérer-recharger-dépenser* », qui supportent d'autres mesures telles que le débit, la latence, l'équité, etc., et qui n'utilisent aucune technique d'estimation pour ajuster le duty-cycle des nœuds dans le réseau.

PARTIE II
CONTRIBUTIONS

Comparaison de la conservation d'énergie des protocoles MAC asynchrones pour les RCSF

4.1 Introduction

La technique de « *duty-cycling* » donne lieu à un défi de communication dans le système, à savoir les émetteurs ne savent pas si les récepteurs sont éveillés et prêts à recevoir le trafic ou dormir pour économiser l'énergie et la recharge. Traditionnellement, les protocoles *MAC* pour les *RCSF* utilisant cette technique peuvent être regroupés en deux catégories: les synchrones et les asynchrones. L'approche de synchronisation est intuitivement inappropriée pour les *RCSF-RE*, car elle ne peut pas prendre en charge les cycles d'activité (*duty-cycle*) individuels indispensables pour adapter la consommation d'énergie à l'énergie ambiante disponible pour chaque nœud individuellement. Considérez que même si le système est réglé de manière à ce que le nœud ayant le moins accès à l'énergie environnementale fonctionne dans l'état *ENO*, les autres nœuds fonctionneront dans des états non optimaux où une partie de l'énergie récoltée est gaspillée. Des exemples de ces protocoles basés sur cette approche sont détaillés dans la section 2.3.2.1 du chapitre 2. Par contre, les approches asynchrones peuvent être classées en deux catégories, à savoir les approches de préambule et de balisage détaillées dans la section 2.3.2.3 du chapitre 2. Les deux catégories asynchrones peuvent efficacement prendre en charge des *duty-cycles* individuels, ce qui est une condition essentielle pour atteindre l'objectif fondamental des *RCSF-RE*, à savoir le fonctionnement adaptatif dans un état qui maximise les performances tout en maintenant un fonctionnement durable. Par conséquent, il devient intéressant d'identifier dans quelles conditions le préambule ou l'approche de balisage est plus appropriée. Ce chapitre se concentre sur la comparaison de la conservation d'énergie des trois protocoles *MAC* asynchrones [54]. Les deux premiers basés sur l'écoute à faible consommation (*LPL*), telle que *BMAC* et *XMAC*, et la seconde sur le paradigme de balisage tel que *RIMAC* en débutant par la définition de la technique de la simulation et par la suite nous présentons le modèle de la consommation d'énergie de chaque protocole et nous détaillons la simulation ainsi que les paramètres utilisés. Ensuite, les résultats des simulations sont discutés et à la fin une conclusion est donnée.

4.2 La simulation

La simulation est une technique rapide, facile, peu coûteuse et souvent très utile pour analyser un système, en particulier dans le domaine de plus en plus complexe et des réseaux de communication. Il est important de noter que la simulation peut aider à prédire les performances du système pendant la phase de conception. La plupart des systèmes informatiques sont analysés via une simulation qui fournit au système de nombreuses

autres alternatives à choisir et à comparer. De plus, le système peut facilement être analysé dans une grande variété de charges de travail et d'environnements par le biais de simulations. Cependant, il est nécessaire de bien comprendre les valeurs de différents paramètres et hypothèses pour un système de simulation, car celle-ci peut produire des résultats prématurés ou trompeurs [122]. De plus, un système représenté par une simulation n'est qu'un modèle abstrait, qui fournit généralement des caractéristiques sélectionnées et spécifiques dans un environnement ou scénario idéal [123].

Dans ce chapitre, nos expériences sont basées sur la plate-forme de simulation de réseau *OMNeT++* en utilisant le modèle de simulation *MiXiM* et les détails de mise en œuvre sont fournis dans ce chapitre où les performances de conservation d'énergie sont évaluées. Les résultats de la simulation montrent que *RIMAC* économise plus d'énergie que *XMAC* et *BMAC*, tandis que le *XMAC* se classe en deuxième position devant *BMAC* (chose qui était prévisible comme indiquée dans le tableau 2.1, mais que nous avons choisi de vérifier). Une brève introduction de *OMNeT++* et du modèle de simulation *MiXiM* est donnée en annexe A.

4.3 Les modèles de la consommation d'énergie

Les observations sur le trafic dans les *RCSF* ont une incidence sur la conception du protocole *MAC*, ainsi que sur les couches réseau et transport. Au lieu d'optimiser pour obtenir un débit élevé, une faible latence et l'équité, les protocoles *MAC* pour les *RCSF* doivent avant tout être éco-énergétiques. Par conséquent, ils doivent être optimisés pour le cas où le trafic réseau est faible ou inexistant pour atténuer les temps morts d'écoute, qui représente la principale cause de la consommation d'énergie [125]. Dans cette section, nous analysons trois protocoles *MAC* asynchrones (*BMAC*, *XMAC* et *RIMAC*) pour les *RCSF* détaillés dans la section 2.3.2.3 du chapitre 2, en termes d'efficacité énergétique. Nous supposons que nous avons un réseau de n nœuds et considérons un type de génération de trafic périodique, où les nœuds génèrent des données avec un intervalle périodique constant. Nous déterminons ensuite les modèles de la consommation d'énergie attendus pour chacun de ces trois protocoles cités.

4.3.1 Le modèle de la consommation d'énergie du protocole *BMAC*

Dans *BMAC*, lorsqu'un nœud i doit transmettre un paquet de données (*DP*), il détecte d'abord la porteuse à chaque intervalle de réveil à un instant T_{cs} représenté dans l'équation (4.2). Si le canal est libre, il transmet un paquet de préambule long (*LPP*) avant chaque *DP*. Il détecte également l'environnement pour mesurer les valeurs physiques à un instant T_{es} présenté dans l'équation (4.3). La consommation d'énergie attendue d'un nœud i est la somme de la consommation d'énergie attendue dans chaque état: détection de la porteuse, détection d'environnement, transmission, réception et veille.

$$\begin{aligned} E_{iB} &= E_{cs} + E_{es} + E_{tx} + E_{rx} + E_{sleep} \\ &= T_{cs}P_{cs} + T_{es}P_{es} + T_{tx}P_{tx} + T_{rx}P_{rx} + T_{sleep}P_{sleep} \end{aligned} \quad (4.1)$$

$$T_{cs} = T_{acs}R_{data} \quad (4.2)$$

$$T_{es} = T_{asense}R_{es} \quad (4.3)$$

Chapitre 4 : Comparaison de la conservation d'énergie des protocoles MAC asynchrones pour les RCSF

Le temps de transmission du nœud i noté par T_{tx} représenté dans l'équation (4.4) est la somme du temps nécessaire pour envoyer un LPP , un DP et un accusé de réception (ACK). Chaque fois que le nœud envoie un DP , il bascule immédiatement sa radio en mode écoute afin de recevoir l'accusé de réception (ACK).

$$T_{tx} = R_{data}(L_p + L_{packet} + L_{ack})T_{(t/r)xb} + 2T_{sw} \quad (4.4)$$

Un nœud peut recevoir m paquets pendant une période donnée de m nœuds appartenant à ses voisins, quoiqu'un seul paquet soit destiné à lui.

$$T_{rx} = R_{data}(mL_p + L_{packet} + L_{ack})T_{(t/r)xb} + T_{sw} \quad (4.5)$$

Enfin, si le nœud ne fait rien, il va dormir pour le reste du temps pendant T_{sleep} .

$$T_{sleep} = 1 - T_{cs} - T_{rx} - T_{tx} - T_{es} \quad (4.6)$$

Symbole	Signification
E_{cs}	Energie de détection de la porteuse
E_{es}	Energie de détection d'environnement
E_{rx}	Energie de réception
E_{tx}	Energie de transmission
E_{sleep}	Energie de veille
P_{cs}	Puissance de détection de la porteuse
P_{es}	Puissance de détection d'environnement
P_{rx}	Puissance de réception
P_{tx}	Puissance de transmission
P_{sleep}	Puissance de veille
R_{data}	Taux de RX/TX d'un paquet de données
R_{es}	Taux pour détecter l'environnement
L_p	Langueur du paquet préambule
L_{sp}	Langueur du paquet préambule court
L_{packet}	Langueur du paquet de données
L_{ack}	Langueur de l'accusé de réception de données
L_{beacon}	Langueur de la balise de <i>Beacon</i>
$L_{beaconR}$	Langueur de la balise de <i>Beacon</i> sur demande (<i>Beacon-on-Request</i>)
$L_{abeacon}$	Langueur de l'accusé de réception de données (<i>Ack_Beacon</i>)
T_{acs}	Temps moyen de la détection de porteuse
T_{cs}	Temps pour détecter un échantillon de radio
T_{es}	Temps de détection d'environnement
T_{asense}	Temps moyen pour échantillonner les données
T_{rx}	Temps de réception
T_{tx}	Temps de transmission
T_{sleep}	Temps de veille
$T_{(t/r)xb}$	Temps de Tx/Rx par octet
T_{sw}	Temps de commutation radio

Tableau 4.1: Les termes utilisés pour les modèles de réseau et d'énergie

4.3.2 Le modèle de la consommation d'énergie du protocole XMAC

Deuxièmement, dans XMAC, qui représente une amélioration de BMAC, des préambules courts (SP) sont envoyés avant chaque paquet de données (DP). La consommation d'énergie prévue par un nœud i est la somme de la consommation d'énergie attendue dans chaque état: détection de la porteuse, détection d'environnement, transmission, réception et veille:

$$\begin{aligned} E_{iX} &= E_{cs} + E_{es} + E_{tx} + E_{rx} + E_{sleep} \\ &= T_{cs}P_{cs} + T_{es}P_{es} + T_{tx}P_{tx} + T_{rx}P_{rx} + T_{sleep}P_{sleep} \end{aligned} \quad (4.7)$$

Comme dans BMAC, un nœud détecte la porteuse avant d'envoyer un SP dans un temps T_{cs} indiqué dans l'équation (4.2). Il détecte également l'environnement dans le temps T_{es} indiqué dans l'équation (4.3).

Le temps de transmission du nœud i noté par T_{tx} représenté dans l'équation (4.8) est la somme du temps nécessaire pour envoyer les paquets suivants: DP, SP et les accusés de réception ACKK et ACK de SP et DP respectivement. Chaque fois qu'un expéditeur doit envoyer un DP, il envoie un SP à un moment précis et modifie directement sa radio en mode écoute afin de recevoir l'accusé de réception (ACKK). Ce processus est répété jusqu'à ce qu'il reçoive un ACKK du nœud cible. K est le nombre de tentatives qu'un nœud envoie un SP et bascule sa radio pour recevoir un ACKK. Après la réception du DP, le récepteur bascule sa radio pour envoyer un ACK.

$$T_{tx} = R_{data}(KL_{sp} + L_{packet} + 2L_{ack})T_{(t/r)xb} + (K + 2)T_{sw} \quad (4.8)$$

Un nœud peut recevoir plusieurs préambules (Σ) au cours d'une période donnée. Néanmoins, lorsqu'il devient un nœud cible pour un préambule spécifique, il envoie immédiatement un ACKK à l'expéditeur et reçoit son DP, puis envoie un ACK à ce dernier.

$$T_{rx} = R_{data}(\Sigma L_{sp} + L_{packet} + 2L_{ack})T_{(t/r)xb} + 2T_{sw} \quad (4.9)$$

Pour calculer le temps de veille du nœud i , on utilise l'équation (4.6) avec les différentes valeurs de durées utilisées, respectivement T_{tx} , T_{rx} , T_{cs} et T_{es} , qui apparaissent respectivement dans les équations (4.8), (4.9), (4.2) et (4.3).

4.3.3 Le modèle de la consommation d'énergie du protocole RIMAC

Enfin, la consommation d'énergie attendue d'un nœud i dans le protocole RIMAC est la somme de la consommation d'énergie attendue dans chaque état: détection de la porteuse, détection d'environnement, transmission, réception et veille.

$$\begin{aligned} E_{iRI} &= E_{cs} + E_{es} + E_{tx} + E_{rx} + E_{sleep} \\ &= T_{cs}P_{cs} + T_{es}P_{es} + T_{tx}P_{tx} + T_{rx}P_{rx} + T_{sleep}P_{sleep} \end{aligned} \quad (4.10)$$

À chaque période de réveil, un nœud récepteur détecte la porteuse avant de transmettre une balise (B) et le nœud émetteur attend un B dans un intervalle de temps à partir du nœud récepteur et détecte également l'environnement, comme le montre

l'équation (4.3). Dans *RIMAC*, il est possible que le nœud émetteur se réveille et détecte la porteuse avant de transmettre une balise sur demande (*BR : Beacon-on-Request*) avant d'émettre un *DP*.

$$T_{cs} \leq 2T_{acs}R_{data} \quad (4.11)$$

Le temps de transmission du nœud est la somme du temps nécessaire pour envoyer une balise *B*, un *DP*, un accusé de réception (*ACKB*) et $W^*(BR)$, où $W \in \{0,1\}$. Un *ACKB* représente, d'une part, la confirmation de réception d'un ancien *DP* envoyé et, d'autre part, une demande de démarrage de la prochaine transmission d'un autre *DP* vers ce nœud. Par conséquent, si l'expéditeur a un nombre N_p de *DP* en attente de transmission alors après la réception de chaque *ACKB*, il envoie un *DP* et change immédiatement sa radio en mode d'écoute afin de recevoir un autre *ACKB*; il répète ce processus jusqu'à l'épuisement de sa file d'attente. Finalement, le nœud émetteur peut envoyer des $Q^*(DP)$ au même récepteur avec ($1 \leq Q \leq N_p$).

$$T_{tx} = R_{data}(L_{beacon} + QL_{packet} + (Q + W)L_{abeacon} + WL_{beaconR})T_{(t/r)xb} + (Q + W + 1)T_{sw} \quad (4.12)$$

Un nœud peut recevoir *B*, $Q^*(DP)$, $m^*(BR)$ et $(Q + W)^*(ACKB)$ pendant une période de temps où ($W \leq m \leq n$) et $W \in \{0,1\}$. D'autre part, lorsqu'il devient un nœud cible pour un *BR* spécifique, il transmet directement un *ACKB* à l'expéditeur et reçoit son *DP*, puis envoie un *ACKB* à ce dernier.

$$T_{rx} = R_{data}(L_{beacon} + QL_{packet} + (Q + W)L_{abeacon} + mL_{beaconR})T_{(t/r)xb} + (Q + W)T_{sw} \quad (4.13)$$

Pour calculer le temps de veille du nœud *i*, nous utilisons l'équation (4.6) avec les différentes valeurs de durées utilisées, respectivement T_{tx} , T_{rx} , T_{cs} et T_{es} , qui apparaissent respectivement dans les équations (4.12), (4.13), (4.11) et (4.3).

4.4 Les outils de simulation et l'implémentation détaillée

Dans cette section, nous présentons les détails de l'implémentation de deux protocoles *XMAC* et *RIMAC* à l'aide de la plate-forme de simulation *OMNeT++-4.6* avec le cadre de simulation *MiXiM-2.3* [8].

WirelessNodeBattery est l'identifiant d'un nœud de réseau existant dans *OMNeT++/MiXiM*, constitué d'un ensemble de modules spécifiques d'un nœud de capteur, tels que *BatteryStats*, qui calcule la consommation d'énergie de chaque module, l'interface réseau *WirelessNicBattery* et l'alimentation *SimpleBattery* (voir la figure A.2 en Annexe A). L'implémentation principale du protocole *XMAC* est *XMacLayer* et pour *RIMAC* est *RIMacLayer*, qui représentent deux modules différents de la couche *MAC* intégrés dans deux nouvelles interfaces réseau, *NicXMAC* pour *XMAC* et *NicRIMAC* pour *RIMAC*. Deux nouveaux hôtes *HostXMAC*, *HostRIMAC* sont créés où chacun de ces hôtes a une interface réseau définie, comme illustré à la figure 4.1(a) et (b) respectivement. La création d'un réseau pour *HostXMAC* et pour *HostRIMAC* et la variation des paramètres

de chaque réseau constituent deux méthodes utiles pour évaluer les performances de *XMAC* et de *RIMAC*.

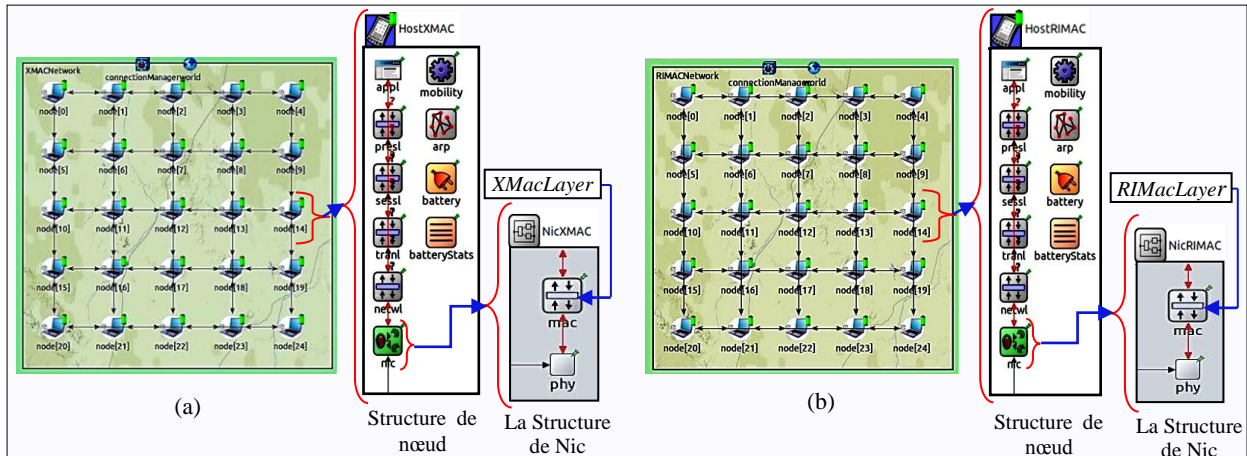


Figure 4.1: (a) un réseau de *XMAC*, (b) un réseau de *RIMAC*

Comme avec les autres modules de *MiXiM*, les deux modules *XMacLayer* et *RIMacLayer* contiennent des portes pour se connecter à d'autres modules ou d'autres couches. Les messages sont utilisés non seulement pour transférer des paquets entre les couches d'un nœud de capteur ou entre deux nœuds, mais également en tant qu'événements pour modifier l'état d'un nœud. Les machines à états finis d'un nœud émetteur et d'un nœud récepteur sont illustrées à la figure 4.2 pour *HostXMAC* et à la figure 4.3 pour *HostRIMAC* qui vont être détaillées par la suite dans cette section. Ces figures illustrent les auto-messages (*selfmessages*) et les paquets réseau utilisés dans *XMacLayer* et dans *RIMacLayer* pour faire pivoter l'état du nœud *HostXMAC* et du nœud *HostRIMAC* respectivement.

4.4.1 La machine à états finis du nœud émetteur et récepteur avec *XMAC*

Après initialisation dans l'état *INIT*, le nœud passe à l'état de veille (*SLEEP*).

Pour le nœud émetteur. Ce dernier, possédant un paquet à transmettre, il passera d'abord dans l'état *CCA* à partir de l'état *SLEEP*, si aucun message n'est détecté au cours de *CCA* (*CCA_timeout* est atteint), il passe à l'état *SEND_PREAMBLE* et commence à émettre un paquet de préambule court comprenant l'adresse du destinataire souhaité. Après chaque transmission de préambule court, il passe à l'état *WAIT_ACKK* si le délai de transmission du préambule court n'est pas atteint (*StopPreambles*); sinon retourne à l'état *SLEEP*. Le nœud passera à *SEND_DATA* après avoir reçu un *ACK* du préambule court (*ACKK*). Une fois que l'expéditeur a transmis le paquet de données au nœud récepteur si le canal est libre, il passe immédiatement à l'état *WAIT_ACK*.

Enfin, pendant l'état *WAIT_ACK* si le nœud reçoit (un accusé de réception de données (*ACK*)) ou (*ACK_timeout* est atteint et $TxAttempts > MaxAttempts$), il revient immédiatement à l'état *SLEEP*; sinon, si (*ACK_timeout* est atteint et que $TxAttempts < MaxAttempts$), il retourne à l'état *SEND_PREAMBLE*.

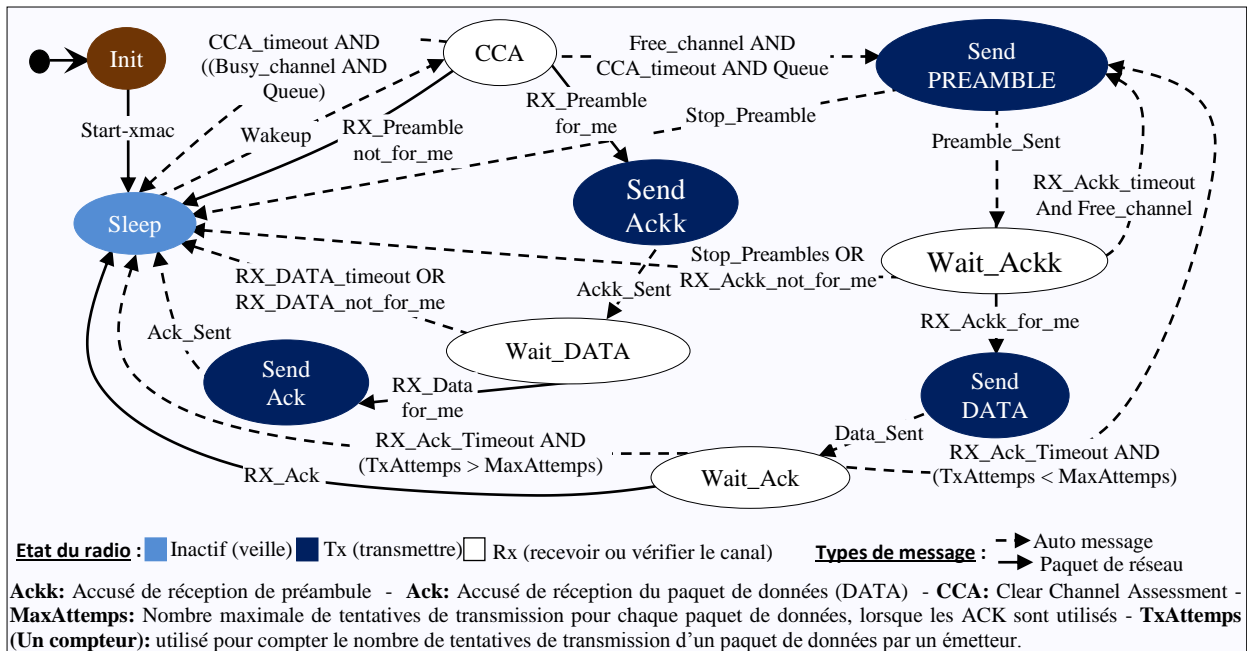


Figure 4.2: Machine à états finis de HostXMAC

Pour le nœud récepteur. Le nœud se réveille pour détecter le canal en vue d'une écoute périodique en passant de l'état *SLEEP* à l'état *CCA*. Si le nœud reçoit un préambule comprenant son adresse de son voisin lors d'un *CCA*, il passe à l'état *SEND_ACKK* pour envoyer *ACKK* au nœud émetteur conforme et passe immédiatement à l'état *WAIT_DATA*; sinon, il retourne à l'état *SLEEP*. Si aucun paquet de données n'est reçu du nœud émetteur conforme pendant *WAIT_DATA*, le nœud passe à l'état *SLEEP*; sinon, il retourne à l'état *SEND_ACK*. Une fois que le destinataire envoie un accusé de réception au nœud émetteur, il passe à l'état *SLEEP*.

4.4.2 La machine à états finis du nœud émetteur et récepteur avec RIMAC

Après l'initialisation dans l'état *INIT*, le nœud passe à l'état *SLEEP*.

1. Premièrement: la transmission de données initiée par le récepteur (la transition entre les états est indiquée par la couleur noire).

Pour le nœud émetteur. Ce dernier ayant un paquet de données à envoyer passera d'abord à l'état *WAIT_BEACON* à partir de l'état *SLEEP*, si aucune communication n'est détectée au cours de cet état (*BEACON_timeout* est atteint), il passe à l'état *SLEEP*; sinon, si le nœud reçoit la balise, il passe à l'état *SEND_DATA*. Une fois que l'expéditeur envoie le paquet de données au nœud récepteur après l'expiration de la période de son *back off*, il passe immédiatement à l'état *WAIT_BEACON_ACK*. Enfin, si (le nœud reçoit un *Beacon_Ack* et que sa file d'attente n'est pas vide) ou (*BEACON_Ack_timeout* est atteint et (*TxAttempts* < *MaxAttempts*)) et que le canal est libre, il passe à l'état *SEND_DATA* sinon, si (*BEACON_Ack_timeout* est atteint et (*TxAttempts* > *MaxAttempts*)) ou (le nœud reçoit un *Beacon_Ack* et sa file d'attente est vide), il retourne à l'état *SLEEP*.

Pour le nœud récepteur. Le nœud se réveille pour détecter le canal en vue d'une écoute périodique en passant d'un état *SLEEP* à un état *CCA*. Si aucune communication n'est détectée pendant *CCA* (*CCA_timeout* est atteint), il passe à l'état *SEND_BEACON* et commence à diffuser une balise de *Beacon* aux nœuds émetteurs voisins et passe à l'état *WAIT_DATA* sinon, il retourne à l'état *SLEEP*. Pendant *WAIT_DATA*, si le nœud reçoit un paquet de données du nœud émetteur correspondant, il passe à l'état *SEND_BEACON_ACK* sinon, si *DATA_timeout* est atteint, il retourne à l'état *SLEEP*. Enfin, une fois que le nœud envoie un *Beacon_Ack* au nœud émetteur, il retourne immédiatement à l'état *WAIT_DATA*.

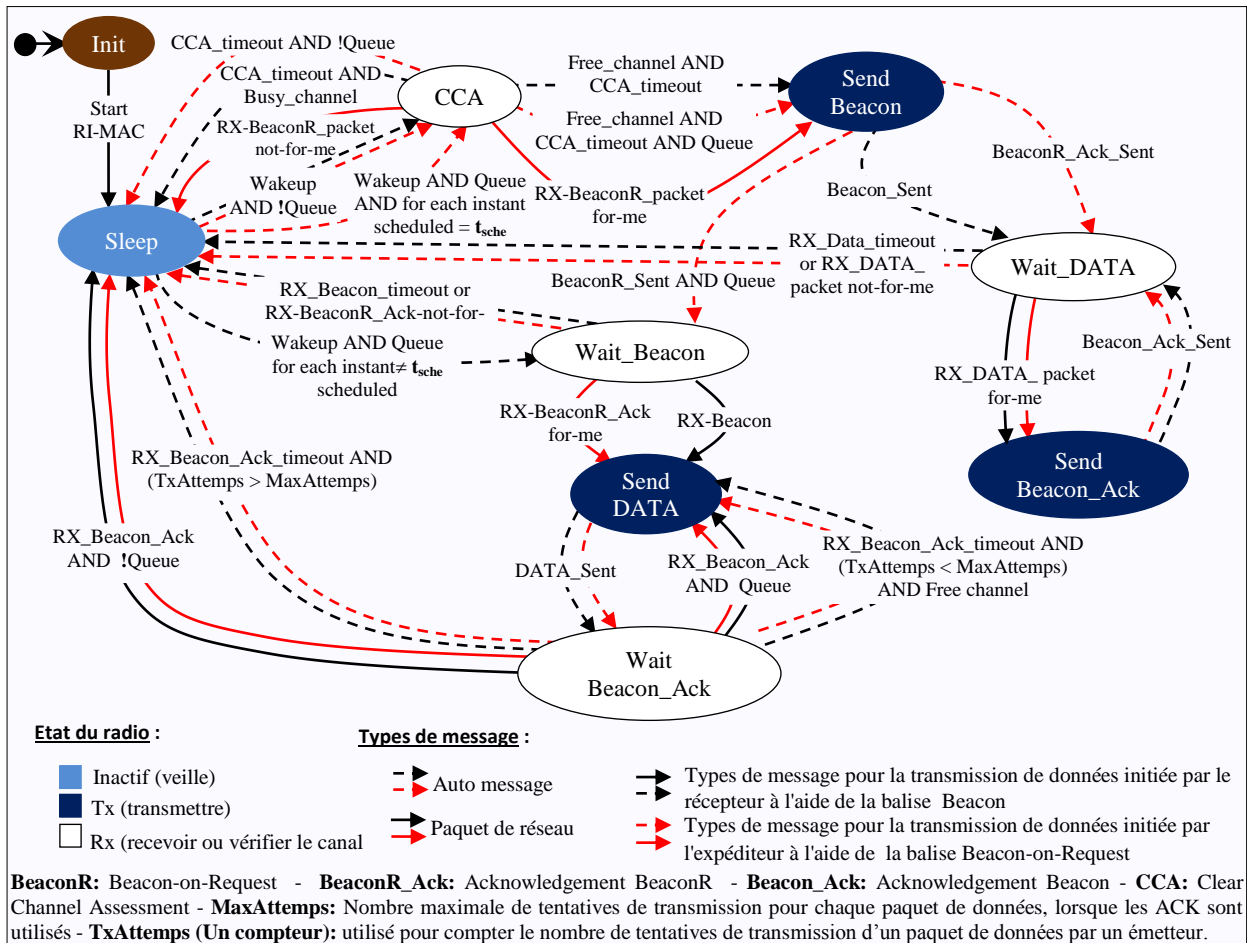


Figure 4.3: Machine à états finis de *HostRIMAC*

2. deuxièmement: la transmission de données déclenchée par l'expéditeur (la transition entre les états est indiquée par la couleur rouge).

Pour le nœud émetteur. Ce dernier ayant un paquet à envoyer passera d'abord à l'état *CCA* à partir de l'état *SLEEP*, si aucune communication n'est détectée pendant *CCA* (le délai *CCA* est atteint), il passe à *SEND_BEACON* et il commence à envoyer une balise sur demande (*BeaconR* : *Beacon-on-Request*) contenant l'adresse du destinataire et il passe à l'état *WAIT_BEACON* sinon, il retourne à l'état *SLEEP*. Le nœud passera à l'état *SEND_DATA* après avoir reçu une balise *ACK* (*BeaconR_Ack*) du destinataire correspondant. Une fois que l'expéditeur envoie le paquet de données au nœud récepteur si le canal est libre, il passe automatiquement à l'état *WAIT_BEACON_ACK*

immédiatement. Enfin, pendant l'état *WAIT_BEACON_ACK* si (le nœud reçoit un *Beacon_Ack* et que sa file d'attente n'est pas vide) ou (*BEACON_Ack_timeout* est atteint et (*TxAttempts* < *MaxAttempts*) et que le canal est libre), il passe à l'état *SEND_DATA* sinon, si (*BEACON_Ack_timeout* est atteint et (*TxAttempts* > *MaxAttempts*)) ou (le nœud reçoit un *Beacon_Ack* et sa file d'attente est vide), il retourne à l'état *SLEEP*.

Pour le nœud récepteur. Le nœud se réveille pour détecter le canal en vue d'une écoute périodique en passant d'un état *SLEEP* à un état *CCA*. Si le nœud reçoit une balise sur demande *BeaconR* contenant son adresse d'un expéditeur de ses voisins lors du contrôle du canal à l'état *CCA*, il passe à l'état *SEND_BEACON* pour envoyer un *BeaconR_Ack* au nœud expéditeur correspondant et il passe automatiquement à l'état *WAIT_DATA* sinon, il retourne à l'état *SLEEP*. Si aucun paquet de données n'est reçu du nœud émetteur correspondant pendant l'état *WAIT_DATA*, le nœud passe immédiatement à l'état *SLEEP* sinon, il revient à l'état *SEND_BEACON_ACK*. Une fois que le destinataire envoie une balise *Beacon_Ack* au nœud émetteur, il passe automatiquement à l'état *SLEEP*.

4.5 Les paramètres de simulation

Paramètre	Valeur
Nombre de nœuds	10, 25, 50, 75, 100
Topologie	Grille
Type d'application	Sensor App Layer
Type de réseau	BaseNetwLayer
Nœud de destination	9, 24, 49, 74, 99
Type de trafic	Périodique
Type de réseau	Stationnaire
Taille du playground (x, y)	400 m x 400 m, 600 m x 600 m, 800 m x 800 m, 950 m x 950 m, 1100 m x 1100 m.
Paramètre de trafic (Traffic Param)	0.5 s, 1.5 s, 2.5 s (s: secondes)
Nombre de paquets envoyés	100, 500, 1000
Type de MAC	<i>BMACLayer</i> – <i>XMACLayer</i> – <i>RIMACLayer</i>
MaxAttempts	2
Taille de la file d'attente	2 <i>pkts</i> (packets)
Puissance de transmission	0.05 W
Capacité de la batterie	1000 <i>mAh</i>

Tableau 4.2: Les paramètres de simulation

Afin d'évaluer les trois protocoles *MAC* asynchrones à conservation d'énergie, nous avons effectué plusieurs simulations et comparé l'énergie consommée par *RIMAC* par rapport au protocole *BMAC*, qui est déjà implémenté dans *MiXiM* et le protocole *XMAC*. Les principaux paramètres de simulation sont spécifiés dans le fichier *omnetpp.ini* et sont résumés dans le tableau 4.2.

Il est à noter que nos résultats sont basés sur la simulation de différents types de réseaux, de 10 nœuds déployés dans une zone de 400 m x 400 m à 100 nœuds de capteurs déployés dans une zone de 1100 m x 1100 m où l'espace entre deux voisins est de 100 mètres pendant 500 secondes de simulation. Dans les différents scénarios utilisés et mis en œuvre, le trafic est généré périodiquement en modifiant le paramètre de trafic (Traffic Param) (i.e. intervalle de temps maximal entre deux transmissions de paquets) en seconde (0,5 s, 1,5 s, 2,5 s) et avec un nombre variable de paquets envoyés (100, 500, 1000). L'énergie initiale de chaque nœud de capteur est définie sur 1000 mAh. La métrique de performance utilisée pour comparer ces trois protocoles de conservation d'énergie asynchrones est la suivante:

Énergie consommée moyenne par un nœud: Il s'agit de la moyenne de l'énergie consommée moyenne (*MCE* : *Mean Consumed Energy*) par un nœud notée par (*Avg* (*MCE*)) dans le RCSF représenté dans l'équation (4.14). Cela devrait être réduit.

$$Avg(MCE) = (\sum_{i=1}^{Nombre\ de\ noeuds} MCE_i) / \text{Nombre de noeuds} \quad (4.14)$$

Où

$$MCE_i = (Er_{fi} - Er_{ii}) / \text{Temps simulé} \quad (4.15)$$

Er_{fi} : L'énergie résiduelle finale du nœud i ,

Er_{ii} : L'énergie résiduelle initiale du nœud i .

4.6 Résultats et Discussion

Les figures 4.4, 4.5 et 4.6 représentent les résultats de la simulation obtenus pour l'énergie consommée moyenne par un nœud avec différents scénarios de trafic où le nombre de nœuds est compris entre 10 et 100. Les résultats indiquent que les conservations d'énergie augmentent avec la diminution du nombre des paquets envoyés et aussi avec la diminution de la valeur du paramètre de trafic (Traffic Param).

Tout d'abord, la figure 4.4, lorsque le paramètre de trafic = 0,5 s, montre que la conservation d'énergie par un nœud dans *RIMAC* est la plus efficace dans la plupart des scénarios, par comparaison avec les deux autres protocoles illustrés aux figures 4.4(b) et 4.4(c) lorsque le nombre de paquets envoyés mis à 500 et 1000 respectivement. Par la suite, *XMAC* est le deuxième meilleur qui permet de conserver plus d'énergie que *BMAC*. Sauf dans les cas où le nombre de paquets envoyés = 100 indiqué à la figure 4.4(a), nous remarquons que *XMAC* est classé premier et *BMAC* en second lieu, avec des performances supérieures à celles de *RIMAC* en raison du taux de livraison élevé des paquets. Pour *XMAC*, la figure 4.4(a) montre un gain plus élevé en termes de conservation d'énergie d'environ 24,15 % lorsque le nombre de nœuds = 10 et d'environ 19,47 % lorsque le nombre de nœuds = 100 par rapport à *RIMAC*. De plus, le *BMAC* de la figure 4.4(a) montre une amélioration de conservation d'énergie d'environ 9,88 % lorsque le nombre de nœuds = 10 et d'environ 6,60 % lorsque le nombre de nœuds = 100 est supérieur à *RIMAC* mais ne dépasse pas le *XMAC*.

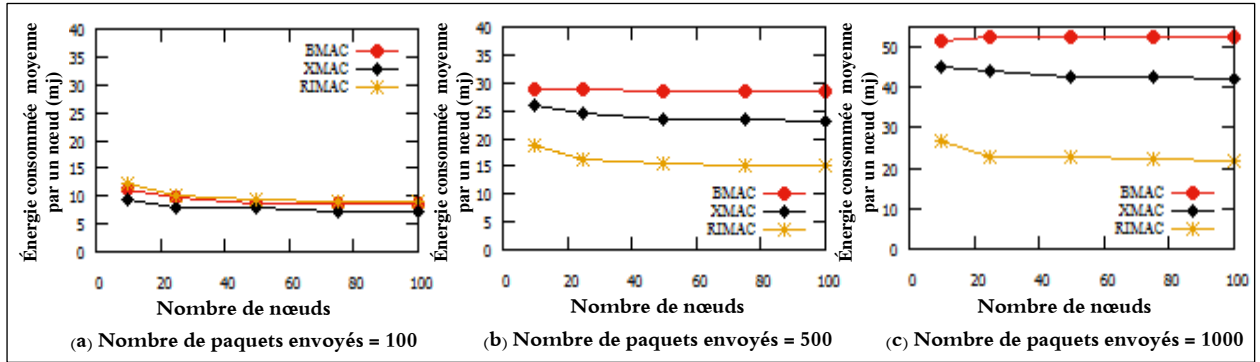


Figure 4.4: Énergie consommée moyenne pour les trois protocoles MAC asynchrones avec un Traffic Param = 0.5 s

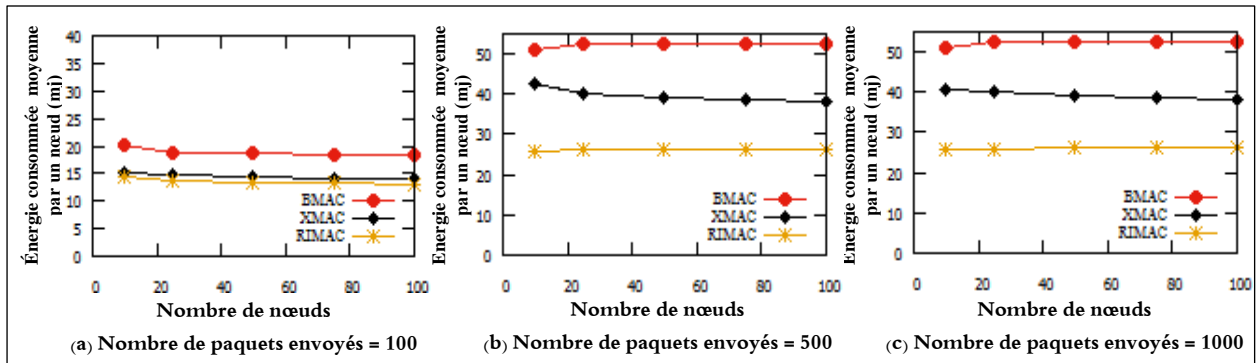


Figure 4.5: Énergie consommée moyenne pour les trois protocoles MAC asynchrones avec un Traffic Param = 1.5 s

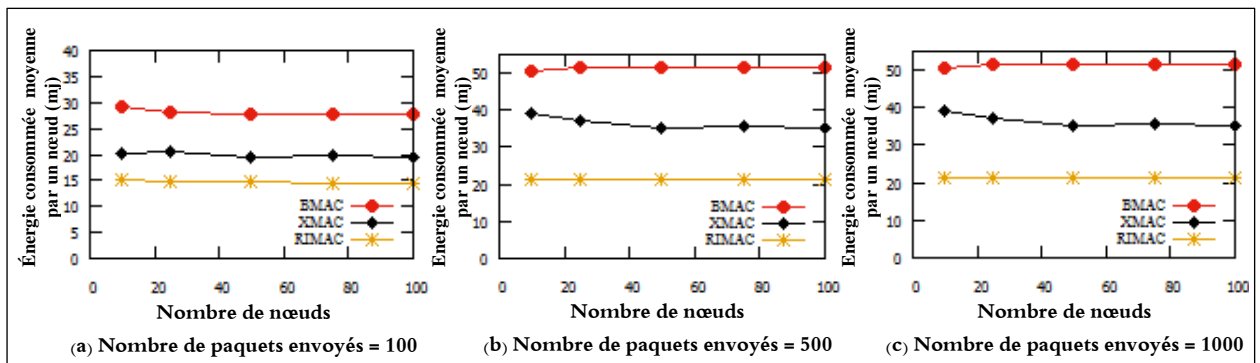


Figure 4.6: Énergie consommée moyenne pour les trois protocoles MAC asynchrones avec un Traffic Param = 2.5 s

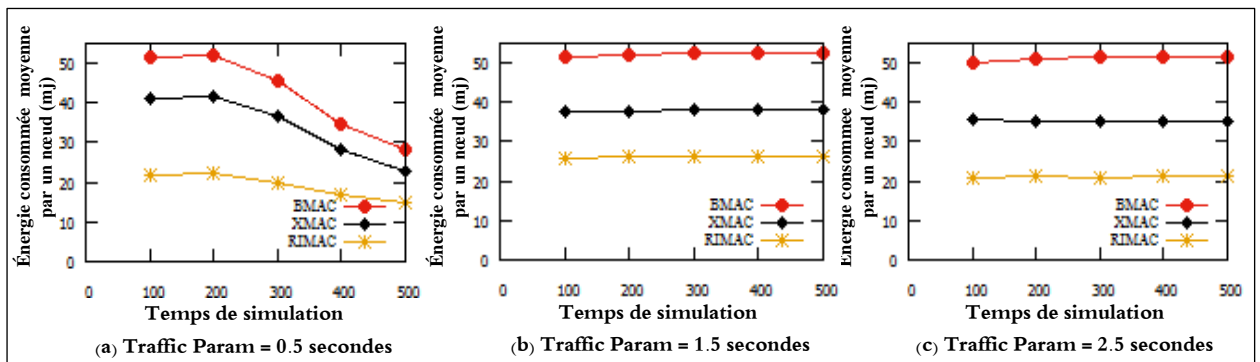


Figure 4.7: Énergie consommée moyenne pour trois protocoles MAC asynchrones avec un nombre de paquets envoyés = 500 et un nombre de nœuds = 100

Cependant, dans les figures 4.4(b) et 4.4(c), *RIMAC* surpasse *BMAC* et *XMAC*. Par exemple, dans la figure 4.4(b), nous remarquons que *RIMAC* a atteint un gain de conservation d'énergie d'environ 28,17 % lorsque le nombre de nœuds = 10 et d'environ 35,02 % lorsque le nombre de nœuds = 100 par rapport à *XMAC*, où il est classé deuxième avant le *BMAC*.

Deuxièmement, les figures 4.5 et 4.6, lorsque le paramètre de trafic est défini sur 1,5 s et 2,5 s respectivement, les résultats montrent que la conservation d'énergie par un nœud dans *RIMAC* sont très efficaces dans tous les scénarios de simulations par rapport à deux autres protocoles. Vient ensuite *XMAC* qui conserve plus d'énergie que *BMAC*. Par exemple, dans la figure 4.5(a), *RIMAC* surpasse le *XMAC* en termes de conservation d'énergie d'environ 4,52 % lorsque le nombre de nœuds = 10 et d'environ 5,69 % lorsque le nombre de nœuds = 100. De plus, les figures 4.5(b) et 4.5(c) illustrent les mêmes résultats que ceux obtenus en termes de conservation d'énergie, où *RIMAC* est classé le premier et *XMAC* le deuxième. Les figures 4.5(b) et 4.5(c) montrent un gain de conservation d'énergie atteint par *RIMAC* d'environ 57,77 %, 45,00 % lorsque le nombre de nœuds = 10 et environ 58,82 %, 39,45 % lorsque le nombre de nœuds = 100 par rapport à *BMAC* et *XMAC*, respectivement. *XMAC* a obtenu un gain de conservation d'énergie d'environ 17,04 % lorsque le nombre de nœuds = 10 et d'environ 27,47 % lorsque le nombre de nœuds = 100 est supérieur à celui de *BMAC*. Pour les figures 4.6(b) et 4.6(c), les mêmes résultats que ceux des figures 4.6(b) et 4.6(c) sont obtenus.

Nous effectuons une autre simulation avec un nombre de nœuds = 100 et un nombre de paquets envoyés = 500 en variant le paramètre de trafic entre (0.5 s, 1.5 s et 2.5 s) pour 5 temps de simulation différents: 100, 200, 300, 400 et 500 secondes. La figure 4.7 montre que l'énergie consommée dans différents scénarios par les protocoles *BMAC*, *XMAC* et *RIMAC*. *RIMAC* conserve plus d'énergie que les deux autres protocoles. Comme d'habitude, le *XMAC* atteint des conservations d'énergie supérieures à celles du *BMAC*. Sur la figure 4.7(a), nous remarquons que les conservations d'énergie diminuent avec l'augmentation du temps de simulation, mais sur les figures 4.7(b) et 4.7(c), la conservation d'énergie se stabilise. Par exemple, dans les figures 4.7(b) et 4.7(c), *RIMAC* surpasse *XMAC* et *BMAC* en termes de conservation d'énergie d'environ 31,48 %, 49,87 % lorsque le temps de simulation = 100 secondes et d'environ 30,82 %, 49,83 % lorsque le temps de simulation = 500 secondes respectivement.

En conclusion, il est utile de souligner que les résultats obtenus sont presque identiques dans tous les scénarios, ce qui montre clairement que le protocole *RIMAC* surpasse les deux autres protocoles *XMAC* et *BMAC* en termes de conservation d'énergie, car ces protocoles génèrent deux coûts d'*over-head*. Premièrement, dans *RIMAC*, l'émetteur consomme de l'énergie en écoutant le canal pour une balise dans chaque transmission de paquets, ce qui est inférieur à sa consommation en transmettant une longue séquence de préambule continue dans *BMAC*, ce qui entraîne une chute de paquets due à une collision ou plusieurs préambules courts dans *XMAC* pour chaque émission de paquet de données. Deuxièmement, dans *RIMAC*, chaque nœud consomme

périodiquement de l'énergie pour transmettre les balises au lieu de consommer de l'énergie fréquemment et inutile dans certains cas d'écouter des préambules sur le canal dans *BMAC* et *XMAC* avant de recevoir le paquet de données.

4.7 Conclusion

Les *RCSF* ont une grande capacité pour différentes applications dans le domaine de la surveillance et du contrôle. Un sujet populaire crucial qui doit être abordé est comment augmenter la durée de vie du réseau en conservant l'énergie des capteurs. L'accent mis sur la conservation d'énergie requiert des solutions spéciales, car les protocoles de communication typiques pour les réseaux sans fil sont conçus pour atteindre un débit élevé, une latence faible et l'équité. Dans ce chapitre, nous avons comparé trois protocoles asynchrones permettant de conserver de l'énergie dans les *RCSF*: *BMAC*, *XMAC* et *RIMAC*. Ces trois protocoles ont en commun d'introduire la technique de duty-cycle pour atténuer les temps morts d'écoute, qui représente la principale cause de la consommation d'énergie. Dans l'ensemble, les résultats des évaluations montrent qu'on peut conclure que *RIMAC* donne de meilleurs résultats par comparaison avec *BMAC* et *XMAC*, tandis que le *XMAC* occupe la deuxième place avant *BMAC*. Le chapitre suivant expose un nouveau schéma *MAC* de gestion de l'énergie dans les *RCSF-RE* qui est basé premièrement; sur le cycle d'activité (duty-cycle) dynamique d'un nœud capteur dans le réseau et deuxièmement; sur la quantité d'énergie récoltée à partir d'une source externe d'énergie ambiante dans le but d'améliorer les performances du *RCSF*. Ainsi, l'amélioration de sa durée de vie.

EEM-EHWSN : Un schéma de gestion de l'énergie amélioré dans les RCSF-RE

5.1 Introduction

La conservation de l'énergie est le principal problème majeur des *RCSF*. En effet, la recharge des sources d'énergie (batteries) dans les réseaux est souvent trop coûteuse, difficile et parfois impossible. Pour prolonger la durée de vie du *RCSF* sans recharger, des méthodes d'économie d'énergie et des systèmes de récupération d'énergie sont essentiels.

Dans les chapitres précédents nous avons présenté et analysé les aspects des différents travaux existants liés premièrement à la gestion de la conservation d'énergie dans les *RCSF* et deuxièmement liés à l'utilisation de différentes formes de la récupération énergétique utilisées dans les *RCSF-RE*. Dans ce chapitre on s'intéresse au stockage et à la gestion de l'énergie récupérée dans les *RCSF* par la proposition d'un schéma de gestion de l'énergie amélioré pour les *RCSF* nommé *EEM-EHWSN* (*Enhanced Energy Management Scheme in Energy Harvesting WSN*) [83]. *EEM-EHWSN* utilise une communication initiée par le récepteur, régule les périodes de veille/réveil en introduisant une politique de seuil d'énergie et en utilisant l'énergie restante afin de réduire le duty-cycle tout en assurant un équilibre entre la consommation d'énergie et la capacité de récupération d'énergie de chaque nœud de capteur dans le *RCSF*.

Nous décrivons dans ce chapitre comment fonctionne notre schéma *EEM-EHWSN*. Dans un premier temps, nous présentons l'architecture du nœud de capteur et le schéma de communication fondamentaux utilisés, puis le modèle de la consommation d'énergie et de la latence. Enfin, nous expliquons comment utiliser la politique de seuil d'énergie proposée pour réguler les périodes de veille et de réveil où la durée de la période de veille est calculée afin de diminuer le duty-cycle.

5.2 Architecture du nœud de capteur utilisée

Dans notre schéma *EEM-EHWSN* proposé, nous avons utilisé un nœud capteur dont l'architecture est de type *Récupérer-recharger-dépenser* détaillée dans la section 3.3.1 du chapitre 3. En utilisant un récupérateur d'énergie, le *RCSF* peut fonctionner en permanence jusqu'à la défaillance matérielle. Chaque nœud de capteur de récupération d'énergie dans *RCSF* comprend généralement un ou plusieurs récupérateurs d'énergie convertissant l'énergie ambiante en énergie électrique afin d'alimenter les autres composants d'un nœud capteur, un dispositif de stockage d'énergie pour stocker l'énergie

recupérée, un capteur de mesure, un processeur pour le traitement, une mémoire pour la sauvegarde des données et un émetteur-récepteur sans fil (*Transeiver*) pour les communications. Les principaux composants d'un nœud d'un RCSF-RE utilisé dans EEM-EHWSN sont illustrés à la figure 5.1.

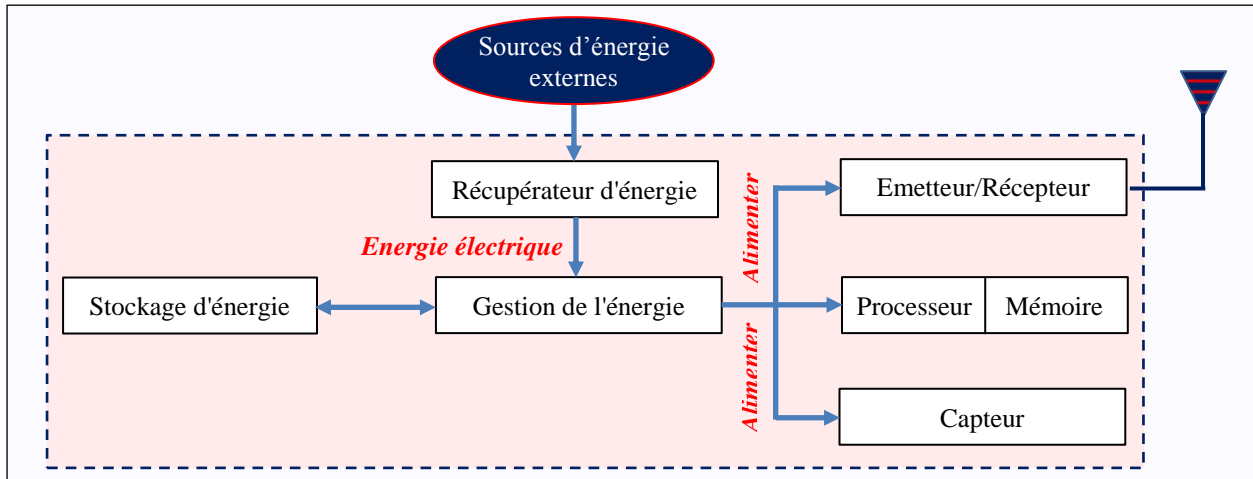


Figure 5.1: Les composants d'un nœud capteur alimenté par un récupérateur d'énergie à partir d'une source externe

5.3 Schéma de communication fondamental utilisé dans EEM-EHWSN

EEM-EHWSN utilise le protocole CSMA/CA pour l'accès au canal, dans lequel un expéditeur utilise une onde porteuse pour rendre le canal occupé et un récepteur écoute le canal sans fil pour éviter les collisions. Dans le schéma EEM-EHWSN, une transmission initiée par le récepteur est utilisée pour éviter le gaspillage d'énergie en l'absence de nœud récepteur prouvé en chapitre 4, chaque nœud récepteur diffuse périodiquement une balise si le canal est libre, ce qui annonce à ses récepteurs qu'il est prêt à accepter la transmission des paquets de données entrants. Tous les nœuds qui possèdent des paquets en file d'attente qui souhaitent être transférés vers le récepteur écoutent le canal en attente d'une balise appropriée. À la réception de la balise, la transmission des paquets de données est initiée.

La figure 5.2 représente la communication de base entre un émetteur et un récepteur dans EEM-EHWSN, comme dans [20] [111]. Dans EEM-EHWSN, chaque nœud se réveille périodiquement en fonction de son propre calendrier pour vérifier s'il existe des trames de données entrantes destinées au nœud en question. Après avoir allumé sa radio, un nœud récepteur *R* émet immédiatement une balise si le canal est libre après la vérification de ce dernier par CCA (*Clear Channel Assessment*), annonçant qu'il est réveillé et prêt à recevoir une trame de données. Un nœud émetteur *E* avec des données en attente à envoyer reste actif silencieusement dans l'attente de la balise du récepteur prévu *R*. A la réception de la balise à partir de *R*, le nœud *E* lance immédiatement sa transmission de données si le délai de son *Back-off* a expiré et le canal est libre qui sera reconnu par *R* avec une autre balise *ACK* (*Acknowledgement*). Notamment, le rôle de la

balise ACK est double: premièrement, il reconnaît la réception correcte de la trame de données envoyée et, deuxièmement, il invite une nouvelle transmission de trame de données au même récepteur. S'il n'y a pas de données entrantes après la diffusion d'une balise ACK, le nœud passe en mode veille.

Dans EEM-EHWSN, il est possible que le nœud émetteur déclenche la communication si l'un des nœuds de son récepteur est actif comme dans [20]. Lorsque l'émetteur E se réveille pour transmettre une trame de données au récepteur R , il envoie d'abord une balise sur demande nommée « *Beacon-on-Request* » après avoir vérifié le canal par CCA, comme indiqué sur la figure 5.3. Dans cette balise, l'expéditeur E définit l'adresse du récepteur R . Si R vient à être actif, il génère une balise en réponse après un délai aléatoire plus long que celui annoncé dans la balise reçue de E . Cette balise générée par le récepteur R permet à l'émetteur E d'émettre des trames de données en attente immédiatement, au lieu d'attendre la balise du nœud R .

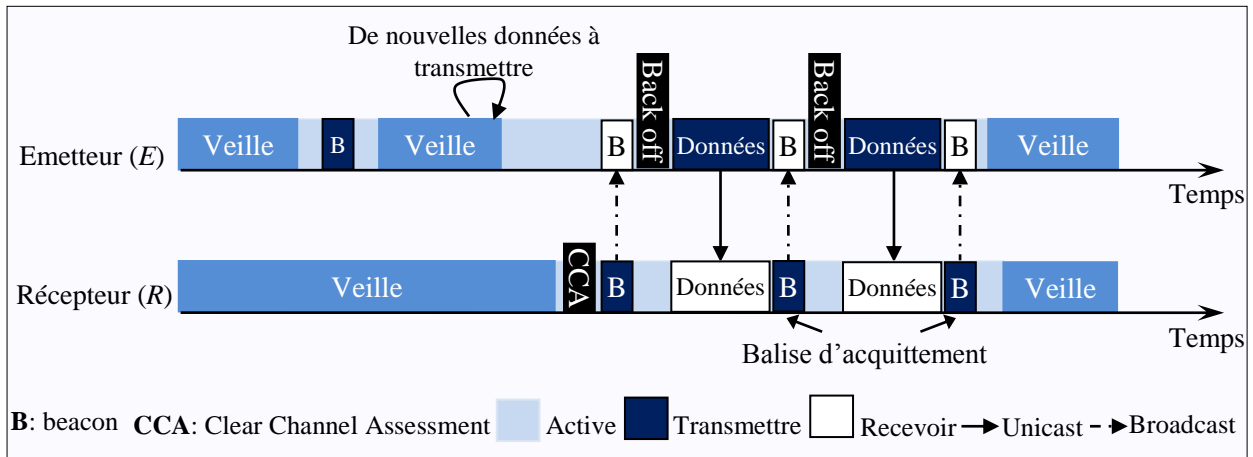


Figure 5.2: Un émetteur E attendra une balise B du récepteur R avant d'envoyer ses données et le double rôle de la balise d'accusé de réception des données dans la communication.

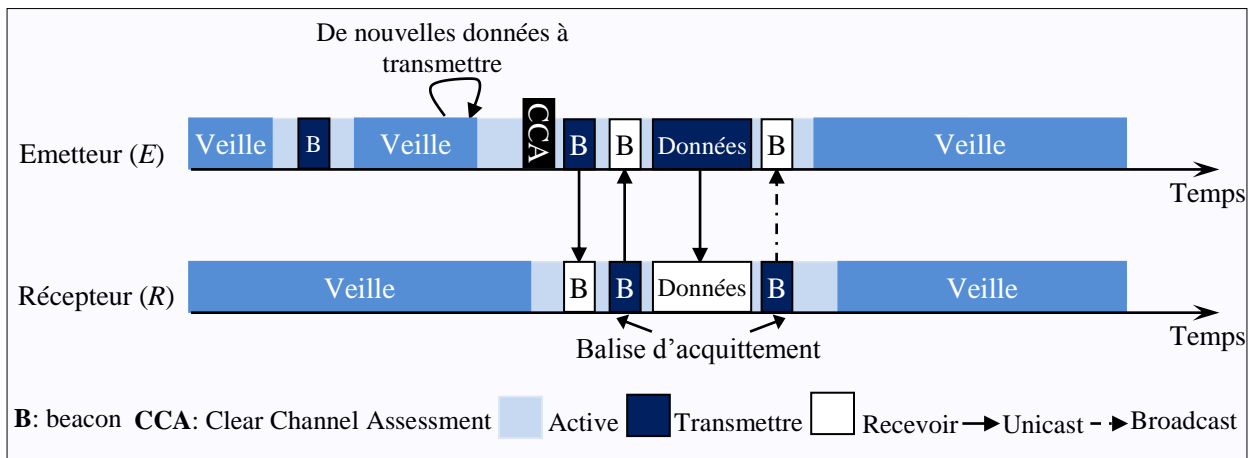


Figure 5.3: Dans EEM-EHWSN, l'émetteur E envoie initialement une balise *Beacon-on-Request* et envoie un paquet de données au nœud R après la réception de la balise d'accusé de réception.

5.4 Le modèle de la consommation d'énergie et de la latence

Nous supposons que nous avons un réseau de K nœuds et que nous considérons un trafic périodique, où les nœuds génèrent des données avec un intervalle périodique fixe. Le tableau 5.1 montre toutes les notations des termes utilisés pour les modèles de réseau, d'énergie et de latence qui seront utilisés dans cette section.

Symbole	Signification
E_{cs}	Énergie de détection de la porteuse
E_{es}	Énergie de détection de l'environnement
E_{rx}	Énergie de réception
E_{tx}	Énergie de transmission
P_{cs}	Puissance de détection de la porteuse
P_{es}	Puissance de détection de l'environnement
P_{rx}	Puissance de réception
P_{tx}	Puissance de transmission
R_{data}	Taux de RX/TX d'un paquet de données
R_{sense}	Taux pour détecter l'environnement
L_{packet}	Langueur du paquet de données
L_{beacon}	Langueur de la balise <i>Beacon</i>
$L_{beaconR}$	Langueur de la balise <i>Beacon-on-Request</i>
$L_{abeacon}$	Langueur de la balise <i>Acknowledgement beacon</i>
T_{cs}	Temps de détection de porteuse
T_{acs}	Temps pour détecter un échantillon de radio
T_{es}	Temps de détection de l'environnement
T_{asense}	Temps moyen pour échantillonner les données
T_{rx}	Temps de réception
T_{tx}	Temps de transmission
$T_{(t/r)xb}$	Temps de T_x/R_x par octet
T_{sw}	Temps de commutation radio
T_{que}	Temps d'attente
T_{pro}	Temps de traitement
n	Nombre des nœuds voisin

Tableau 5.1: Les termes utilisés pour les modèles : réseau, l'énergie et la latence dans EEM-EHWSN

5.4.1 Le modèle de la consommation d'énergie

La consommation d'énergie attendue en état de réveil pendant la période t par un nœud i dans EEM-EHWSN est la somme de la consommation d'énergie attendue dans chaque état suivant: détection de la porteuse, détection d'environnement, transmission et réception.

$$E_i(t) = E_{cs}(t) + E_{es}(t) + E_{tx}(t) + E_{rx}(t) = T_{cs}P_{cs} + T_{es}P_{es} + T_{tx}P_{tx} + T_{rx}P_{rx} \quad (5.1)$$

À chaque intervalle de réveil, un nœud récepteur détecte le canal avant d'envoyer une balise, le nœud émetteur attend une balise dans un intervalle de temps à partir du nœud récepteur et détecte également l'environnement pour mesurer les valeurs physiques. Dans le schéma EEM-EHWSN, il est possible que le nœud émetteur se réveille et détecte le canal avant d'envoyer une balise sur demande « *Beacon-on-Request* » pour transmettre une trame de données.

$$T_{cs} \leq 2T_{acs}R_{data} \quad (5.2)$$

$$T_{es} = T_{asense}R_{sense} \quad (5.3)$$

Le temps de transmission du nœud est la somme du temps requis pour envoyer une balise de *Beacon*, un paquet de données, une balise d'accusé de réception et Z balise sur demande « *Beacon-on-Request* » avec: $Z \in \{0,1\}$. Une balise d'accusé de réception est utilisée à la fois comme un accusé de réception pour des données précédemment reçues et comme une initiation pour demander la transmission de données suivante à ce nœud. En conséquence, si un expéditeur à Np paquets de données à envoyer, après la réception de chaque balise d'acquittement, il envoie un paquet de données et change immédiatement sa radio en mode écoute afin de recevoir une autre balise d'accusé de réception; ce processus est répété jusqu'à ce que la file d'attente de l'expéditeur soit vide. En fin de compte, le nœud émetteur peut transmettre Q paquets de données au même récepteur avec ($1 \leq Q \leq Np$).

$$T_{tx} = R_{data}(L_{beacon} + QL_{packet} + (Q + Z)L_{abeacon} + ZL_{beaconR})T_{(t/r)xb} + (Q + Z + 1)T_{SW} \quad (5.4)$$

Un nœud peut recevoir une balise, Q paquets de données, m balise sur demande « *Beacon-on-Request* » et $(Q + Z)$ balises d'acquittement pendant une période avec : ($Z \leq m \leq n$) et $Z \in \{0,1\}$. Cependant, lorsqu'il un nœud devient un nœud cible pour une balise sur demande « *Beacon-on-Request* » spécifique, il envoie immédiatement un accusé de réception de balise à l'expéditeur et reçoit son paquet, puis envoie une balise d'accusé de réception à ce dernier.

$$T_{rx} = R_{data}(L_{beacon} + QL_{packet} + (Q + Z)L_{abeacon} + mL_{beaconR})T_{(t/r)xb} + (Q + Z)T_{SW} \quad (5.5)$$

5.4.2 Le modèle de la latence

Après la mise en file d'attente et le traitement de chaque paquet dans chaque nœud. Le nœud reçoit une balise ou envoie une balise avant chaque paquet de données et envoie un accusé de réception (balise) après la réception de chaque paquet de données. Selon les équations. (5.2), (5.4) et (5.5), la latence attendue est:

$$\begin{aligned}
 L &= T_{que} + T_{pro} + T_{cs} + T_{tx} + T_{rx} \\
 &= T_{que} + T_{pro} + T_{cs} + 2R_{data}(L_{beacon} + QL_{packet} + (Q + Z)L_{abeacon} + \\
 &\quad ZL_{beaconR})t_{(t/r)xb} + 2(Q + Z + 1/2)T_{SW}
 \end{aligned} \tag{5.6}$$

5.5 La politique de seuil d'énergie proposée dans EEM-EHWSN

EEM-EHWSN admet deux modes de duty-cycle comme dans [111]. Le premier est le mode statique où les paramètres (par exemple, temps de réveil et temps de veille) sont entièrement définis par l'administrateur et le second est le mode dynamique, où les paramètres sont ajustés en fonction de la quantité d'énergie récupérée et de l'énergie restante qui seront utilisés dans le reste de fonctionnement du EEM-EHWSN. Souvent, les nœuds avec une meilleure alimentation en énergie se réveillent plus, avec une période de veille plus courte, et relaient ainsi plus des paquets. De plus, afin de minimiser les collisions résultant des nœuds voisins, qui se réveillent en même temps et se disputent le canal. EEM-EHWSN sélectionne consciemment les durées de veille en fonction de la disponibilité énergétique des nœuds. Il bascule certains nœuds en mode relais pour ne relayer que les données de ses voisins afin d'atteindre l'équilibre de charge dans RCSF. Par conséquent, un nœud avec un niveau d'énergie inférieur sélectionne une période de veille plus longue pour collecter plus d'énergie.

5.5.1 Mécanisme de gestion de l'énergie récupérée pour les RCSF

Pour améliorer les performances du RCSF et sa durée de vie, la gestion de la consommation d'énergie et de la récupération d'énergie est cruciale. Dans cette section, nous décrivons un mécanisme de gestion de l'énergie récupérée proposé appelé EH2M (*Energy-Harvested Management Mechanism for WSN*) pour RCSF, afin de minimiser (optimiser) le duty-cycle sur la base de quantité d'énergie restante. Avec EH2M, chaque nœud de capteur peut ajuster son duty-cycle de manière dynamique en calculant son temps de veille en fonction de l'énergie restante. De plus, notre mécanisme proposé établit un équilibre entre la consommation d'énergie et la capacité de récupération d'énergie de chaque nœud de capteur dans RCSF.

5.5.1.1 Formulation du problème d'optimisation

Cette section introduit la formulation du problème d'optimisation de l'énergie d'un nœud traité par EH2M [84] que nous avons utilisé dans nos contributions (EEM-EHWSN et PS-EHWSN). L'objectif principal est de minimiser le duty-cycle d'un nœud de capteur i dans RCSF soumis à une contrainte de rapport du temps d'activité t_{active_i} requis par ce

dernier, la disponibilité de l'énergie du nœud est limitée par un seuil d'énergie. Nous supposons que nous avons un RCSF de n nœuds de capteurs et considérons un type de collecte d'énergie périodique, où chaque nœud i est composé d'une unité d'alimentation de capacité maximale (E_{max}) et un système de récupération d'énergie. De plus, un seuil d'énergie (E_{th}) est utilisé par chaque nœud afin d'assurer un équilibre entre la consommation d'énergie et la capacité de récupération d'énergie. Le temps de veille t_{sleep_i} d'un nœud i est calculé sur la base de sa quantité d'énergie restante E_{r_i} représentée dans l'équation 5.10, en utilisant l'équation suivante montrée dans la section 5.5.2.2 dont le temps de vérification t_{check} est introduit:

$$t_{sleep_i} = \left(\frac{E_{r_i} - E_{MAX}}{E_{th} - E_{MAX}} \right) \times T_{Ste_i} \quad (5.7)$$

Où : T_{Ste_i} : Temps de veille écoulé par le nœud i jusqu'à ce que son énergie restante E_{r_i} atteigne le seuil d'énergie E_{th} avec :

$$T_{Ste_i} = \min(T_{Ste_i}, 1) \quad (5.8)$$

Afin de minimiser le duty-cycle Dc_i d'un nœud de capteur i , représenté dans l'équation (2.1), un problème d'optimisation (OP) est formulé comme suit :

OP

$$\text{minimiser } Dc_i \quad \forall i \in \{1, \dots, k\} \quad (5.9a)$$

Sous contraintes:

$$t_{active_i} \leq t_{active_i} + t_{sleep_i} \quad (5.9b)$$

$$t_{active_i} + t_{sleep_i} \neq 0 \quad (5.9c)$$

$$E_{th} \leq E_{r_i} \leq E_{MAX} \quad (5.9d)$$

$$E_{th} < E_{MAX} \quad (5.9e)$$

$$t_{active_i}, t_{sleep_i} \geq 0 ; E_{th}, E_{r_i}, E_{MAX} > 0 \quad (5.9f)$$

5.5.1.2 Les fonctions de base du EH2M

Avec EH2M, le duty-cycle d'un nœud est fixé au démarrage où le temps actif et le temps de veille sont déterminés en totalité par l'administrateur et il devient dynamique en fonction du taux auquel l'énergie peut être récoltée une fois que l'énergie restante E_{r_i} est inférieure que E_{th} ($E_{r_i} < E_{th}$) comme le montre la figure 5.4. À plusieurs reprises, les nœuds avec une meilleure alimentation en énergie se réveillent davantage, avec un temps de veille court, et reçoivent ainsi plus de paquets. De plus, ce mécanisme réduit les collisions résultant de nœuds voisins qui se réveillent en même temps et en concurrence sur le canal. Grâce à EH2M, les temps de veille des nœuds sont choisis consciemment en fonction de la disponibilité en énergie des nœuds, qui ne peut probablement pas être identique pour un ensemble de nœuds. Par conséquent, un nœud avec un niveau d'énergie inférieur sélectionne un temps de veille long pour récolter la plus grande énergie possible afin de stocker un niveau d'énergie garanti. À travers EH2M le temps de veille pour le nœud i (t_{sleep_i}) est déterminé par la procédure suivante:

1. à l'état de réveil, chaque nœud i vérifie son énergie restante juste avant de passer à l'état de veille;
2. à l'état de veille, si ($E_{r_i} < E_{th}$), le nœud i calcule le temps de veille écoulé (t_{ste_i}), jusqu'à ce que son énergie restante atteigne E_{th} , en utilisant l'équation 5.8;
3. dans les prochains états de réveil, tant que $E_{r_i} \geq E_{th}$, le nœud i calcule son temps de veille (t_{sleep_i}) en utilisant l'équation 5.7; sinon, le nœud i exécute les étapes (1), (2) puis (3) respectivement.

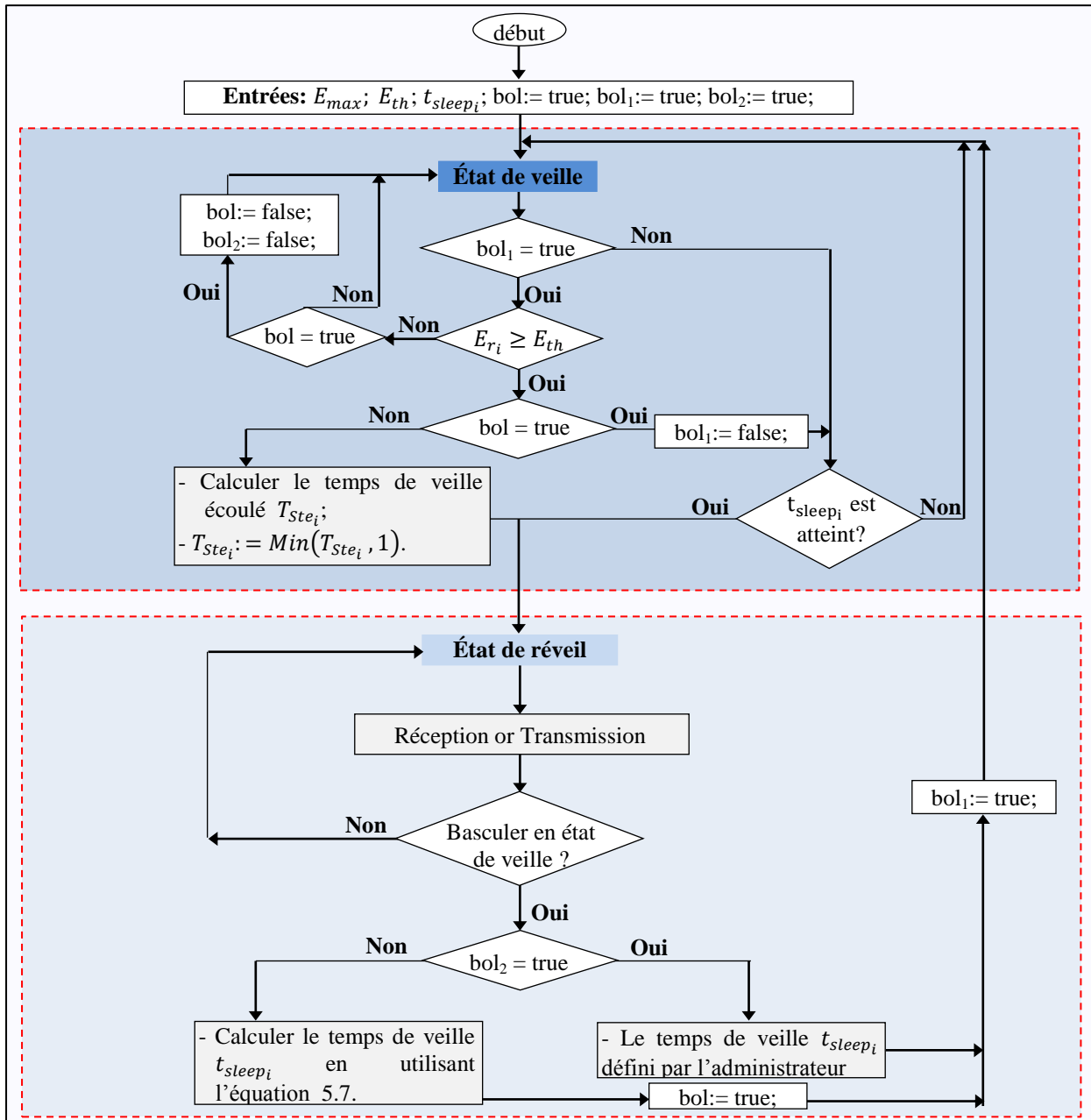


Figure 5.4: Fonctions de base du mécanisme EH2M

5.5.2 Aperçu de la politique de seuil d'énergie

En résumé, la politique de seuil d'énergie proposée dans EEM-EHWSN, pour le calcul de la durée de la période de veille et la régulation de la période active du nœud du capteur, fonctionne en deux étapes:

5.5.2.1 L'étape de la commutation

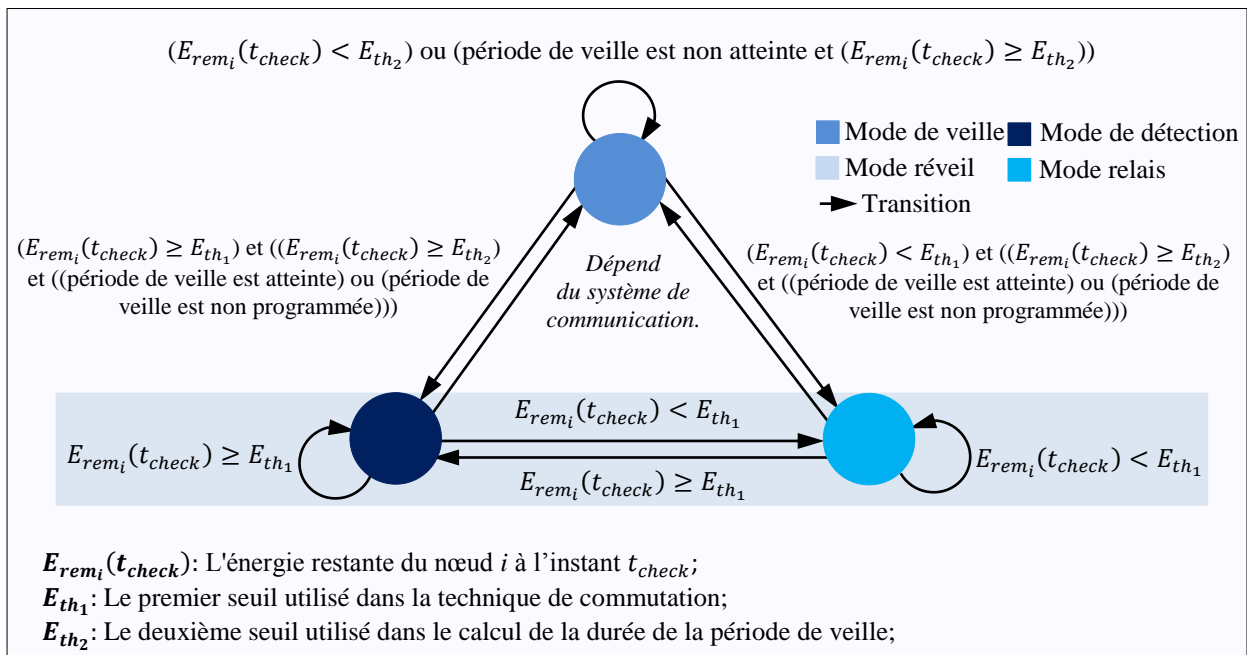


Figure 5.5: Diagramme de transition d'état pour les trois modes possibles du nœud de capteur dans EEM-EHWSN.

L'objectif principal de cette étape est de réguler la période de réveil et d'assurer un équilibre de charge dans le RCSF, où chaque nœud de capteur peut basculer entre trois modes possibles tout en fonctionnant sur le réseau, comme le montre la figure 5.5. Cette procédure dépend du taux d'énergie récupéré et de son impact sur la quantité d'énergie restante. Ces modes sont:

1. **Mode de détection** : dans ce mode de détection, le nœud collecte les données (événements) et les transmet ultérieurement à la station de base (nœud récepteur).
2. **Mode relais** : dans ce mode, le nœud relais ne lie que les données de ses nœuds voisins. Le mode de détection et le mode relais se réalisent en mode (état) de réveil.
3. **Mode (état) veille** : dans ce mode, l'interface radio du nœud est désactivée.

5.5.2.2 L'étape de calcul de la durée de la période de veille

L'objectif principal de cette étape est de réduire le duty-cycle afin de diminuer la quantité de l'énergie résultant de la collision de paquets et aussi, de réduire les contentions de canal et les retransmissions pour un paquet de données dans le but de réduire la latence moyenne du paquet. Pendant la période de réveil et en mode de duty-cycle dynamique, chaque nœud calcule la durée de la période de veille suivante en fonction de l'énergie restante en utilisant notre nouveau mécanisme développé EH2M détaillé dans la section 5.5.1.

Afin de réaliser ces deux étapes de cette politique, chaque nœud de capteur utilise deux seuils d'énergie définis par l'administrateur: le premier est E_{th_1} , qui sera utilisé dans la technique de commutation proposée par tous les nœuds pour modifier leur mode

de fonctionnement (en mode relais ou mode de détection). Le second est E_{th_2} plus proche de E_{th_1} où ($E_{th_2} < E_{th_1}$), qui est un autre seuil d'énergie. Il est utilisé par chaque nœud avec l'énergie restante pour le calcul de la durée de la période de veille en fonction de l'interpolation de Lagrange en état de réveil si l'énergie restante du nœud de capteur i à l'instant t_{check} ($E_{rem_i}(t_{check})$) est supérieure ou égale au seuil E_{th_2} ($E_{rem_i}(t_{check}) \geq E_{th_2}$). Dans cette politique, l'énergie restante à l'instant de vérification t_{check} du nœud i ($E_{rem_i}(t_{check})$) est calculée en utilisant l'équation suivante:

$$E_{rem_i}(t_{check}) = \left[E_{present_i}(t'_{check}) - E_{consumed_i}(t_{check} - t'_{check}) \right] + \phi_i(t_{check} - t'_{check}) \quad (5.10)$$

Où:

$E_{present_i}(t'_{check})$: L'énergie actuelle du nœud i à l'instant de vérification précédent t'_{check} ;

$E_{consumed_i}(t_{check} - t'_{check})$: Est calculée en utilisant l'équation (5.1) et représente l'énergie consommée pendant la période ($t_{check} - t'_{check}$) du nœud i où ($t'_{check} < t_{check}$);

$\phi_i(t_{check} - t'_{check})$: La quantité d'énergie récoltée pendant la période ($t_{check} - t'_{check}$) par le nœud i , où ($t'_{check} < t_{check}$).

5.5.3 Le mécanisme proposé pour le calcul de la période de veille

5.5.3.1 L'utilisation de la technique de commutation proposée en état de réveil

L'idée principale de cette technique est la suivante: en état de réveil, chaque nœud de capteur vérifie périodiquement l'énergie restante dans sa batterie. Si cette dernière à l'instant de vérification t_{check} est inférieure au seuil d'énergie E_{th_1} , le nœud changera son état en mode relais; sinon, il passera en mode de détection comme indiqué en (5.11). Les étapes ci-dessus sont répétées pendant la durée de vie du RCSF.

$$MSSN = \begin{cases} \text{mode de détection si } E_{rem_i}(t_{check}) \geq E_{th_1} \\ \text{mode de relais si } E_{rem_i}(t_{check}) < E_{th_1} \end{cases} \quad (5.11)$$

Où $MSSN$ (*Mode Switching of Sensor Node*): mode de commutation du nœud de capteur dans le réseau.

5.5.3.2 La procédure proposée pour calculer la durée de la période de veille

Les étapes pour calculer la durée de la période de veille (d_{sleep_i}) d'un nœud juste avant qu'il soit en état de veille sont déterminées en mode de duty-cycle dynamique par la procédure proposée. Les différentes étapes de cette procédure se résument comme suit:

1. dans l'état de réveil, si l'énergie restante à l'instant de vérification t_{check} du nœud i est égale à son énergie maximale E_{max} , la durée maximale de la période de veille d_{max} est nulle;
2. en mode de réveil, chaque nœud i vérifie son énergie restante juste avant de passer en mode veille;
3. si l'énergie restante d'un nœud i ; $E_{rem_i}(t_{check})$ est inférieure au seuil d'énergie E_{th_2} juste après qu'il soit en état de veille, il calcule la durée de la période de veille écoulée d_{th_i} jusqu'à ce que son énergie restante atteigne le seuil d'énergie E_{th_2} , en utilisant l'équation suivante:

$$d_{th_i} = \min(t_{cs_i} - t_{is_i}, 1) \quad (5.12)$$

Où :

t_{cs_i} : L'instant de vérification, où $E_{rem_i}(t_{cs_i}) \geq E_{th_2}$, lorsque le nœud i est en état de veille;

t_{is_i} : L'instant de vérification initial, juste après que le nœud i est en état de veille, où $E_{rem_i}(t_{is_i}) < E_{th_2}$.

4. dans les états de réveil suivants, tandis que l'énergie restante a l'instant de la vérification t_{check} d'un nœud i est supérieure ou égale à l'énergie seuil E_{th_2} , la durée de la prochaine période de veille (d_{sleep_i}) est calculée sur la base de l'interpolation de Lagrange (5.14) et les deux paramètres suivants: (E_{max}, d_{max}) et (E_{th_2}, d_{th_i}) sont déterminés par l'exécution des deux instructions précédentes (1) et (3); sinon, le nœud i exécute les étapes (2), (3) puis (4) respectivement.

$$d_{sleep_i}(E_{rem_i}(t_{check})) = \frac{E_{rem_i}(t_{check}) - E_{max}}{E_{th_2} - E_{max}} * d_{th_i} + \frac{E_{rem_i}(t_{check}) - E_{th_2}}{E_{th_2} - E_{max}} * d_{max} \quad (5.13)$$

Puisque d_{max} est nul, la formule (5.13) devient:

$$d_{sleep_i}(E_{rem_i}(t_{check})) = \frac{E_{max} - E_{rem_i}(t_{check})}{E_{max} - E_{th_2}} * d_{th_i} \quad (5.14)$$

5.5.3.3 Les fonctions de base du EEM-EHWSN

Notre schéma proposé (EEM-EHWSN) est basé sur un duty-cycle dynamique, qui assemble les périodes de réveil et les périodes de veille, et offre une chance respectable aux récupérateurs d'énergie de recharger la source d'énergie avant d'entrer dans un état de déchargement où l'énergie déposée est consommée par le nœud de capteur. EEM-EHWSN est basé sur les fonctions de base suivantes, comme indiqué sur la figure 5.6:

1. initialement, chaque nœud de capteur i passe en mode veille au démarrage et après chaque instant de vérification t_{check} , il calcule son énergie restante $E_{remi}(t_{check})$;
2. si l'énergie restante $E_{remi}(t_{check})$ est inférieure au seuil d'énergie E_{th_2} , il initialise son t_{is_i} (l'instant de démarrage de la récupération d'énergie) ; sinon, si t_{is_i} est initialisé, il calcule d'abord son t_{cs_i} et ensuite son d_{th_i} (la durée de la période de veille écoulée par le nœud i jusqu'à ce que son énergie restante atteigne le seuil énergétique E_{th_2}) grâce à l'utilisation de l'équation (5.12) sinon, il vérifie directement la condition pour passer en état de réveil;
3. il passe en état de réveil selon l'une des deux conditions de commutation suivantes:
 - 1- si l'énergie restante $E_{remi}(t_{check})$ est supérieure ou égale au seuil d'énergie E_{th_2} après avoir récolté la quantité d'énergie requise où le d_{th_i} est calculée.
 - 2- si la durée de la période de veille d_{sleep_i} est expirée définie par l'administrateur dans le mode de duty-cycle statique qui ne sera utilisée qu'une seule fois par chaque nœud capteur pendant le fonctionnement du réseau tant que l'énergie restante est supérieure ou égale au seuil d'énergie E_{th_2} , ou il est calculé périodiquement dans le mode de duty-cycle dynamique sur la base de l'équation (5.14) en fonction de la quantité d'énergie récupérée et de l'énergie restante $E_{remi}(t_{check})$ lorsque cette dernière est supérieure ou égale au seuil d'énergie E_{th_2} avant qu'il soit en état de veille, qui sera utilisé dans le reste de fonctionnement du *EEM-EHWSN*;
4. dans l'état de réveil, chaque nœud de capteur à chaque instant de vérification t_{check} vérifie son énergie restante $E_{remi}(t_{check})$. Si cette dernière est inférieure au seuil d'énergie E_{th_1} , il changera son état en mode relais; sinon, il modifiera son état en mode de détection. Les étapes ci-dessus sont répétées pendant la durée de vie du *RCSF*;
5. si la condition de basculement à l'état de veille est vérifiée après l'utilisation du système de communication (réception ou transmission de (*Beacon*, *BeaconR*, *BeaconR_Ack*, paquet de données ou *Beacon_Ack*)) appliqué par le schéma *EEM-EHWSN* utilisé par chaque nœud de capteur i , ce dernier calcule le nouveau prochaine durée de la période de veille d_{sleep_i} grâce à l'utilisation de l'équation (5.14) sur la base de l'énergie restante $E_{remi}(t_{check})$ qui est calculée à l'aide de l'équation (5.10) si la valeur restante de l'énergie $E_{remi}(t_{check})$ est supérieure ou égale au seuil d'énergie E_{th_2} ou il utilise la durée de la période de veille définie par l'administrateur pour le mode de duty-cyle statique ou calculée dans le mode dynamique précédent si d_{th_i} (la durée de la période de veille écoulée par le nœud i jusqu'à ce que son énergie restante $E_{remi}(t_{check})$ atteigne le seuil d'énergie E_{th_2}) n'est pas calculée.

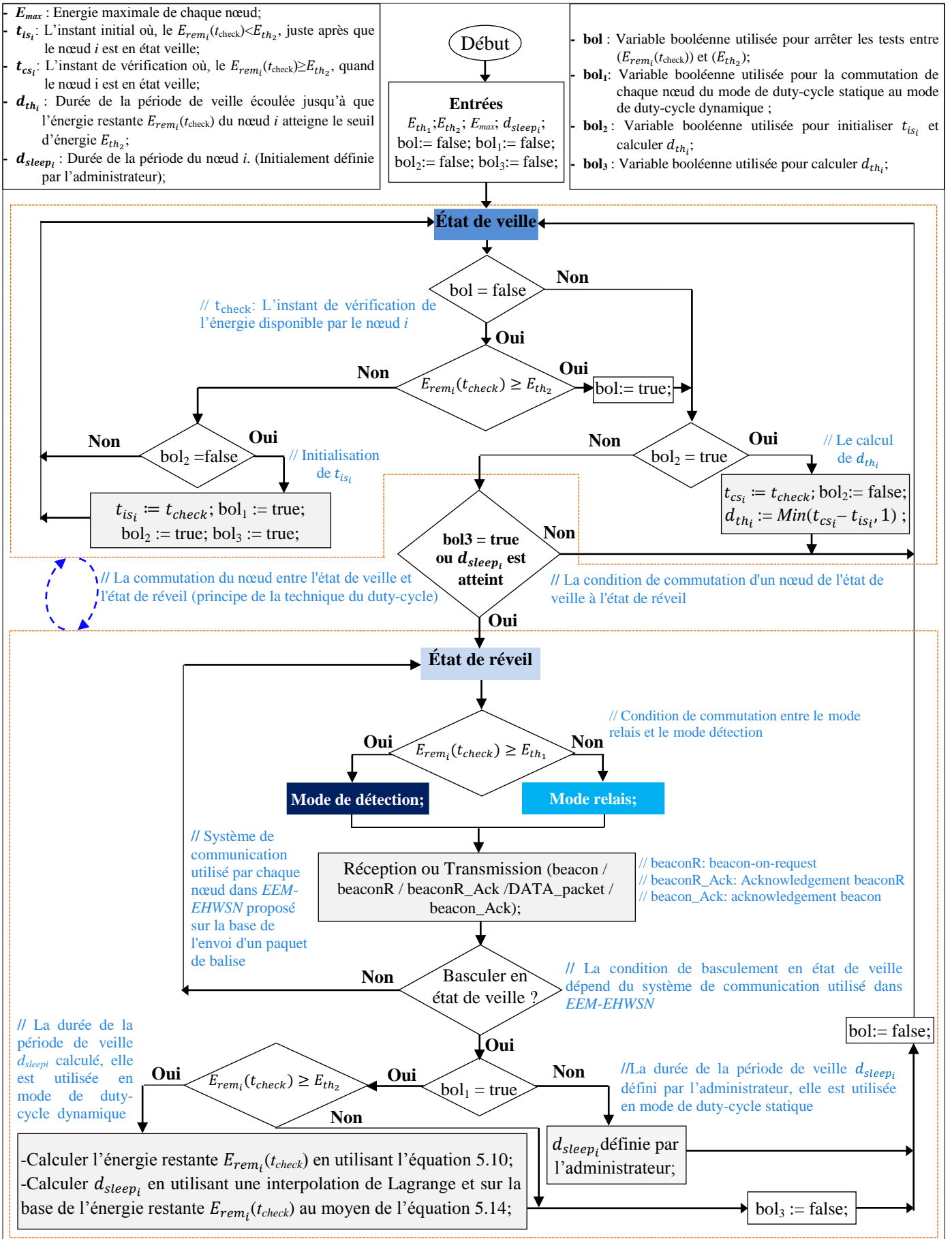


Figure 5.6: Fonctions de base du système EEM-EHWSN où chaque nœud de capteur est équipé d'un système de récupération d'énergie

5.6 La simulation

Cette section présente premièrement la simulation du mécanisme *EH2M* comparé avec *ENO* et deuxièmement la simulation du schéma *EEM-EHWSN* comparé avec le protocole *RIMAC* et les deux schémas *DSR* et *DSP*.

1. Premièrement: la simulation du mécanisme *EH2M*. Nous avons utilisé la plateforme de simulation *OMNeT++-4.6 / MiXiM-2.3* pour implémenté les deux mécanismes de gestion d'énergie *EH2M* et *ENO* intégrés dans le type de communication *MAC (RI)* initié par le récepteur (détaillée dans la section 4.4.2 du chapitre 4), respectivement. Nos résultats sont basés sur la simulation de 16 nœuds de capteurs déployés dans une zone de $400\text{ m} \times 400\text{ m}$ à 100 nœuds de capteurs déployés dans une zone de $1100\text{ m} \times 1100\text{ m}$ dans lesquels la distance entre deux voisins dans la grille est de 100 mètres pendant 500 secondes de temps de simulation. Tous les nœuds de capteur source peuvent envoyer leur trafic périodique toutes les 1,5 secondes vers le *sink* via des transmissions multi-sauts et la période de récupération d'énergie est égale à 0,1 seconde pour chaque nœud de capteur. La quantité d'énergie récoltée est choisie au hasard dans un intervalle prédéfini [$0\text{ mW} \dots 50\text{ mW}$]. L'énergie initiale de chaque nœud de capteur est réglée sur 50 J et la valeur d'énergie maximale est de 100 J avec la valeur du seuil d'énergie définie sur 30 J . Les métriques de performance utilisées sont les suivantes:

- 1. Taux de livraison des paquets:** le nombre de paquets reçus à la destination, divisé par le nombre de paquets envoyés depuis les sources. Cela devrait être maximisé.
- 2. Latence moyenne des paquets:** le temps nécessaire au paquet pour atteindre le nœud de destination est calculé en moyenne pour tous les paquets. Cela devrait être minimisé.
- 3. Débit:** le taux de transfert total (octets reçus/durée de la simulation) qui devrait être maximisé.

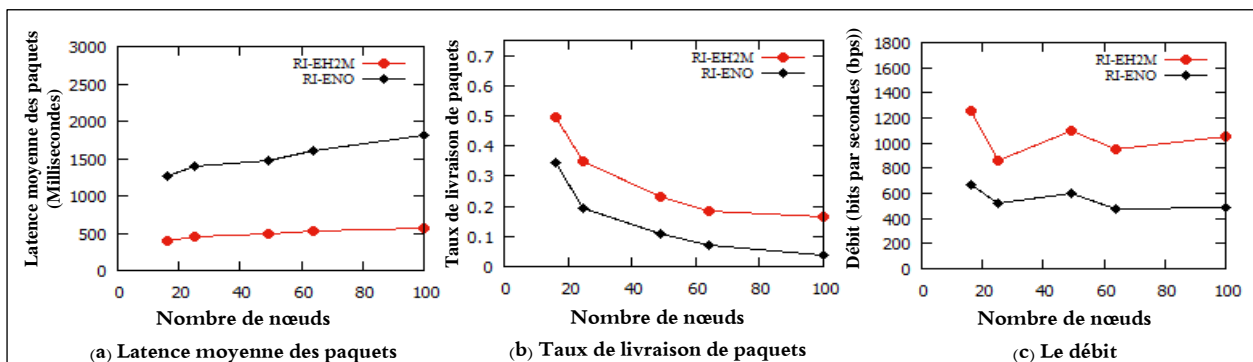


Figure 5.7: Les métriques de performance. (a) Latence moyenne, (b) Taux de livraison de paquets, (c) Le Débit

La figure 5.7(a) représente les résultats de la simulation pour la métrique de latence moyenne avec une valeur de trafic périodique égale à 1,5 secondes et un nombre de paquets envoyés égal à 100, le nombre de nœuds compris entre 16 et 100. Les résultats montrent que *RI-EH2M* surpasse considérablement *RI-ENO*. Il est considérable que la latence moyenne de *RI-EH2M* est inférieure à celle de *RI-ENO*, car *RI-EH2M* minimise plus le duty-cycle, minimisant ainsi le temps d'activité. À l'inverse, dans *RI-ENO*, le

duty-cycle n'est pas minimisé mieux, ce qui augmente le temps d'activité, ce qui provoque la contention locale des demandes de transmission ainsi que la retransmission des messages de contrôle avant l'envoi du paquet de données. Cette retransmission due à une latence moyenne élevée qui est traitée dans notre mécanisme *EH2M* par un ajustement du duty-cycle dans *RCSF* par le calcul du temps de veille de chaque nœud de capteur sur la base de la quantité d'énergie restante. En raison de la variabilité de la consommation d'énergie et de la récupération d'énergie entre les nœuds, il est moins probable que les nœuds du voisinage se réveillent en même temps. Pour *RI-EH2M*, la figure 5.7(a) montre un gain en termes de latence moyenne inférieure d'environ 68,70 % par rapport à *RI-ENO* lorsque le nombre de nœuds est égal à 16 et d'environ 68,52 % lorsque la taille du réseau est définie sur 100.

La figure 5.7(b) présente les résultats obtenus pour la métrique du taux de livraison de paquets. Nous observons que le taux de livraison de paquets diminue lorsque le nombre de nœuds augmente. Les résultats montrent que *RI-EH2M* surpasse *RI-ENO* lorsque le nombre de nœuds augmente pour un taux variable de récupération d'énergie. Ce résultat est dû au fait que chaque nœud de capteur peut ajuster son duty-cycle de manière dynamique sur la base de l'énergie restante lorsque les durées de veille sont choisies en connaissance de cause, ce qui ne risque pas d'être identique pour un voisinage, minimisant ainsi la collision et les retransmissions. Par exemple, lorsque le nombre de nœuds passe de 16 à 49, le taux de livraison de paquets diminue de 53,17 % pour *RI-EH2M* et de 68,45 % pour *RI-ENO*. En outre, on peut remarquer sur cette figure que des gains de performance significatifs d'environ 75,99 % du taux de livraison des paquets ont été obtenus de *RI-EH2M* plus que *RI-ENO*, lorsque le nombre de nœuds = 100.

La figure 5.7(c) représente les résultats de la simulation obtenus pour la métrique de débit calculée en bits par secondes (*bps*). Les résultats montrent que *RI-EH2M* est capable de fournir un débit supérieur à celui de *RI-ENO* grâce à la réduction dynamique de la période active des nœuds de capteurs, qui dépend de la quantité d'énergie récupérée. Cela minimise la collision et le nombre de retransmissions, car plus il y a de collisions, plus il faut de temps pour qu'une transmission réussie entraîne la perte de paquets. D'autre part, *RI-ENO* fournit un faible débit de paquets en raison de l'utilisation d'une période active statique dans la plupart de temps de nœuds de capteurs et n'exploite pas l'énergie future disponible pour ajuster le duty-cycle des nœuds de capteurs. Pour *RI-EH2M*, la figure 5.7(c) montre un gain en termes de débit élevé d'environ 46,86 % par rapport à *RI-ENO* lorsque le nombre de nœuds est égal à 16 et d'environ 53,33 % lorsque la taille du réseau est définie sur 100.

Globalement, on peut constater que les gains de performances significatifs en termes de latence moyenne, de taux de livraison de paquets et de débit sont produits par *EH2M* plus que *ENO*.

2. Deuxièmement: la simulation du schéma EEM-EHWSN. Dans la section suivante nous présentons l'implémentation détaillée de notre schéma *EEM-EHWSN* qui basé sur la technique de commutation et sur l'utilisation du mécanisme *EH2M*.

5.6.1 Les outils de simulation et les détails de l'implémentation

Pour implémenter notre schéma *EEM-EHWSN*, nous avons utilisé la plate-forme de simulation *OMNeT++* (version 4.6) avec le modèle de simulation *MiXiM* (version 2.3). Dans cette section, les détails de l'implémentation du *EEM-EHWSN* sont expliqués.

WirelessNodeBattery est l'identifiant d'un nœud de réseau existant dans *OMNeT++/ MiXiM*, composé d'un ensemble de modules spécifiques d'un nœud de capteur tels que *BatteryStats*, qui calcule la consommation d'énergie de chaque module, l'interface réseau *WirelessNicBattery* et l'alimentation *SimpleBattery*.

La principale implémentation du schéma *EEM-EHWSN* est *EEMEHWSNLayer*, qui un module de la couche *MAC* inclut dans une nouvelle interface réseau *NicEEMEHWSN* basée sur *WirelessNicBattery* (voir la section 4.4 du chapitre 4). Un nouvel hôte *HostEEMEHWSN* (basé sur le module de récupération d'énergie que nous avons implémenté, chargé de récupérer la quantité d'énergie selon l'un des types suivants: périodique, uniforme ou exponentiel; et sur *WirelessNodeBattery* qui est composé d'autres couches différentes telles que : la couche application, présentation, session, transport, réseau et *NIC* (composé de deux couches : *MAC* et *physique*)) est créé et contiennent cette interface réseau, comme indiqué sur la figure 5.8. La création d'un réseau *HostEEMEHWSN* et la modification des paramètres réseau constituent deux méthodes utiles pour évaluer les performances du *EEM-EHWSN*. Tout comme les autres modules de *MiXiM*, *EEMEHWSNLayer* est constitué de portes pour connecter avec d'autres modules ou couches. Les messages sont manipulés non seulement en tant que paquets à envoyer entre les couches d'un nœud de capteur ou entre deux nœuds, mais également en tant qu'événements pour modifier l'état d'un nœud.

EEMEHWSNLayer utilise quatre fonctions de message de traitement importantes pour fonctionner; la première s'appelle *handleUpperMsg*, qui gère l'envoi du *DATA_packet* de la couche supérieure (couche réseau) via l'appel de la fonction *sendDown*. Après avoir reçu ce *DATA_packet* par le nœud récepteur, un signal est envoyé à la couche physique située en dessous afin d'activer l'antenne radio (si est en mode veille et le temps de veille est expiré) ou bascule en état de réception; la seconde s'appelle *handleLowerMsg*, qui est utilisée pour manipuler les paquets de contrôles (c'est-à-dire *Beacon*, *Beacon_Ack*, *BeaconR_Ack* pour le nœud émetteur) qui sont envoyés depuis la couche inférieure (couche physique) via l'appel de la fonction *sendUp*; la troisième s'appelle *handleSelfMsg*, responsable de la gestion de l'état de la couche *MAC*, en fonction du type des auto-messages reçus utilisés comme temporisateurs dans le réseau (*EEMEHWSNNetwork*) et la quatrième fonction est appelée *handleLowerControl*, chargée de gérer le contrôle de message contenant les informations d'état de l'antenne radio, telles que la transmission d'un paquet est terminée ou que l'antenne radio commute sur l'état de réception ou de transmission est terminé. Ce type de message de contrôle est échangé entre *EEMEHWSNLayer* et la couche physique lors de leur communication via une autre porte.

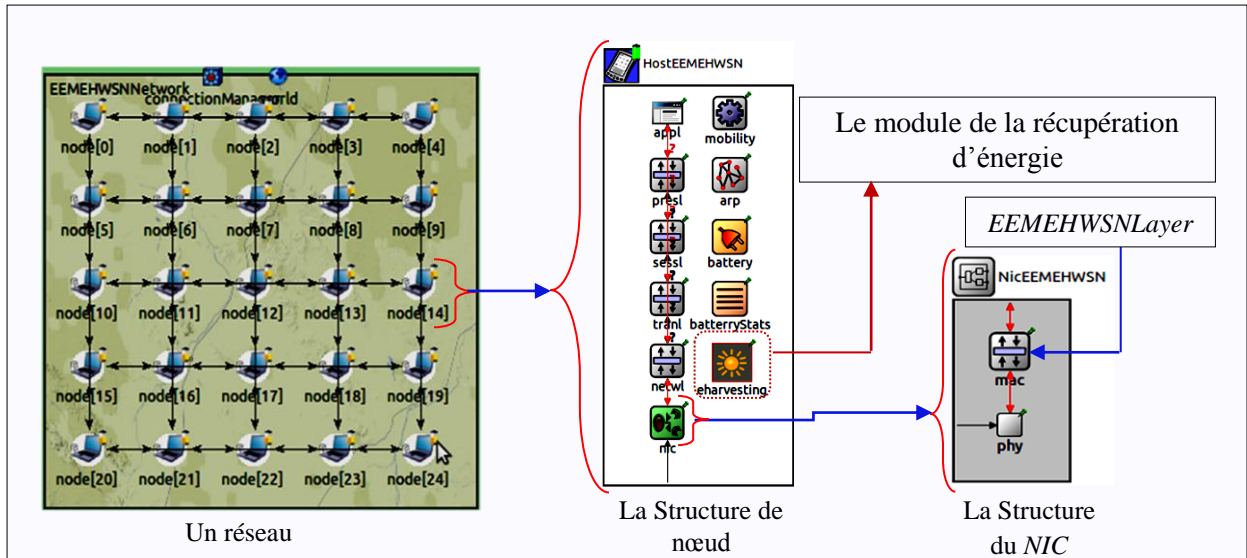


Figure 5.8: Le réseau du schéma EEM-EHWSN

5.6.1.1 La machine à états finis du nœud émetteur et récepteur avec EEM-EHWSN

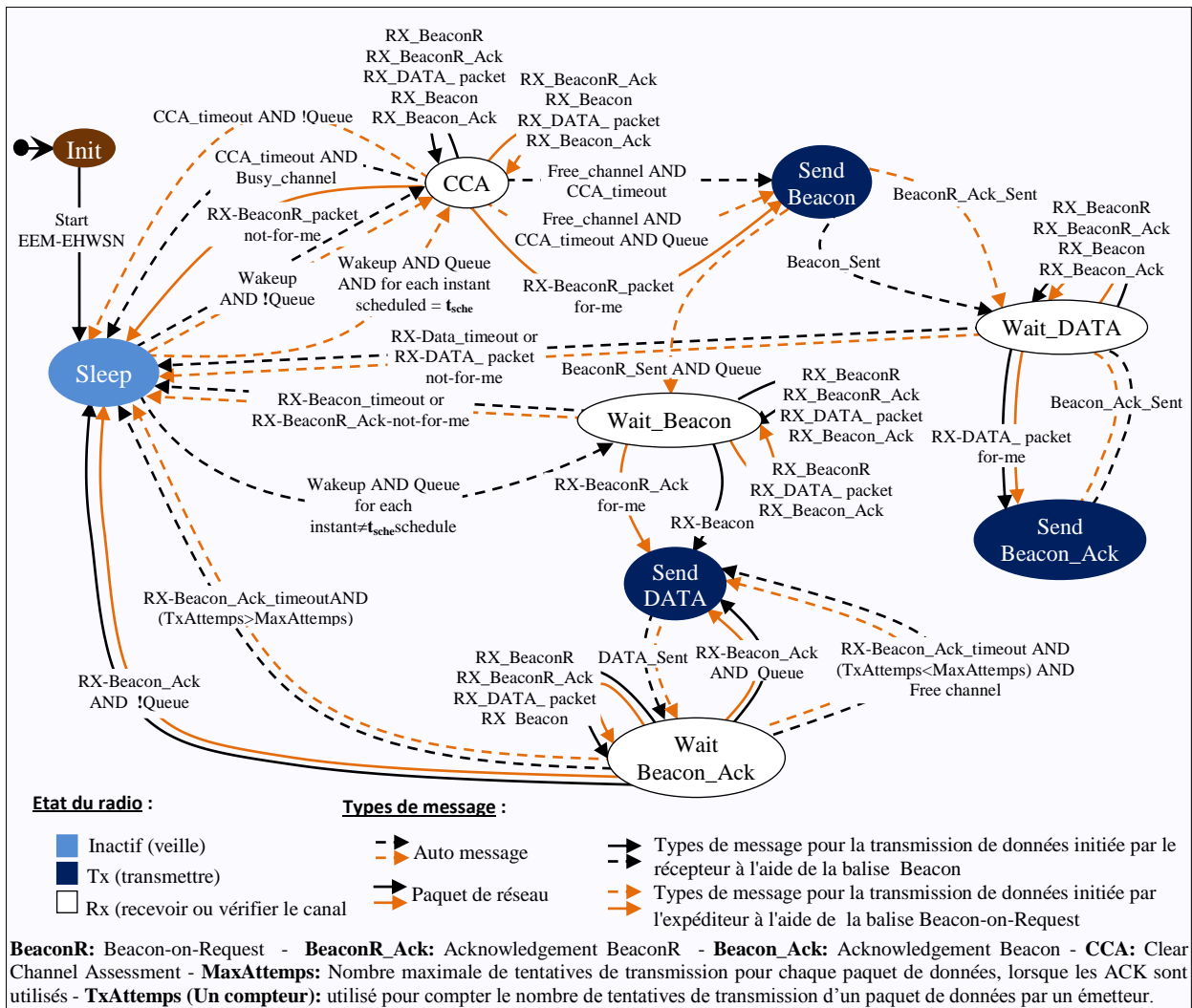


Figure 5.9: Machine à états finis de HostEEMEHWSN

Dans cette section on présente la description de la machine à états finis du nœud émetteur et récepteur illustrée à la figure 5.9 qui illustre tous les auto-messages (*self-messages*) et les paquets du réseau utilisés dans *EEMEHWSNLayer* pour modifier l'état d'un nœud *HostEEMEHWSN*. Après initialisation dans *INIT*, le nœud passe à l'état de veille (*SLEEP*).

1. Premièrement: la transmission de données initiée par le récepteur (la transition entre les états est indiquée par une couleur noire).

Pour le nœud émetteur. Un émetteur ayant un paquet de données (*DATA*) à envoyer passe d'abord à l'état *WAIT_BEACON* à partir de l'état *SLEEP*; si aucune communication n'est détectée pendant cet état (*Beacon_timeout* est atteint), il passe à l'état *SLEEP*; sinon, si le nœud reçoit une balise *Beacon*, il passe à l'état *SEND_DATA*; sinon, s'il reçoit un *DATA_packet* ou *Beacon_Ack* ou *Beacon_R* ou *BeaconR_Ack*, il reste dans l'état actuel. Une fois que le *DATA_packet* est transmis au nœud récepteur après l'expiration de la période de *back-off*, il passe automatiquement à l'état *WAIT_BEACON_ACK*. Enfin, si (le nœud reçoit un *Beacon_Ack* et que sa file d'attente n'est pas vide) ou (*Beacon_Ack_timeout* est atteint et (*TxAttempts* < *MaxAttempts*)) et que le canal est libre), il passe à l'état *SEND_DATA*; sinon, si (*Beacon_Ack_timeout* est atteint et (*TxAttempts* > *MaxAttempts*)) ou (le nœud reçoit un *Beacon_Ack* et que sa file d'attente est vide), il retourne à l'état *SLEEP*; sinon, si le nœud reçoit une balise *Beacon* ou un paquet de données ou *Beacon_R* ou *BeaconR_Ack*, il reste dans l'état actuel.

Pour le nœud récepteur. Le nœud se réveille pour détecter le canal en vue d'une écoute périodique en passant de l'état *SLEEP* à l'état *CCA*. Si le nœud reçoit une balise *Beacon* ou *Beacon_Ack* ou *DATA_packet* ou *Beacon_R* ou *BeaconR_Ack*, il reste dans l'état actuel; sinon, si aucune communication n'est détectée pendant *CCA* (*CCA_timeout* est atteint), il passe à l'état *SEND_BEACON* et commence à diffuser une balise *Beacon* aux nœuds émetteurs voisins. Ensuite, il passe à l'état *WAIT_DATA*; sinon, il retourne à l'état *SLEEP*. Pendant *WAIT_DATA*, si le nœud reçoit un *DATA_packet* d'un nœud émetteur correspondant, il passe à l'état *SEND_BEACON_ACK*; sinon, si *Data_timeout* est atteint, il retourne à l'état *SLEEP*; sinon, s'il reçoit une balise *Beacon* ou *Beacon_Ack* ou *Beacon_R* ou *BeaconR_Ack*, il reste dans l'état actuel. Enfin, une fois que le nœud envoie un *Beacon_Ack* au nœud émetteur, il passe immédiatement à l'état *WAIT_DATA*.

2. Deuxièmement: la transmission de données initiée par l'expéditeur (la transition entre les états est colorée en orange).

Pour le nœud émetteur. Ce dernier ayant un paquet à envoyer passera d'abord à l'état *CCA* à partir de l'état *SLEEP*, si le nœud reçoit une balise *Beacon* ou *Beacon_Ack* ou *DATA_packet* ou *BeaconR_Ack*, alors il reste dans l'état actuel; sinon, si aucune communication n'est détectée pendant le *CCA* (*CCA_timeout* est atteint), il passe à *SEND_BEACON* et commence à envoyer une (*Beacon-on-request* notée par *BeaconR*) contenant l'adresse du destinataire prévu, puis il passe à l'état *WAIT_BEACON*; sinon il retourne à l'état *SLEEP*. Pendant *WAIT_BEACON*, si le nœud reçoit une balise

Beacon_Ack, un *DATA_packet* ou *BeaconR*, il reste dans l'état actuel; sinon, il passera à *SEND_DATA* après avoir reçu une *Beacon_Ack* du destinataire correspondant. Une fois que l'expéditeur envoie *DATA_packet* au nœud récepteur si le canal est libre, passe automatiquement à l'état *WAIT_BEACON_ACK*. Enfin, pendant l'état *WAIT_BEACON_ACK* si (le nœud reçoit un *Beacon_Ack* et que sa file d'attente n'est pas vide) ou (*Beacon_Ack_timeout* est atteint et ($TxAttempts < MaxAttempts$) et que le canal est libre), il passe à l'état *SEND_DATA*; sinon, si (*Beacon_Ack_timeout* est atteint et ($TxAttempts > MaxAttempts$)) ou (le nœud reçoit *Beacon_Ack* et que sa file d'attente est vide), il retourne à l'état *SLEEP*; sinon, il reste dans l'état actuel.

Pour le nœud récepteur. Le nœud se réveille pour détecter le canal en vue d'une écoute périodique en passant de l'état *SLEEP* à l'état *CCA*. Si le nœud reçoit *Beacon* ou *Beacon_Ack* ou un paquet de données (*DATA_packet*) ou *BeaconR_Ack* (accusé de réception de *BeaconR*), il reste dans l'état actuel; sinon, s'il reçoit *Beacon-on-Request* contenant son adresse de son expéditeur voisin pendant le *CCA*, il passe à l'état *SEND_BEACON* pour envoyer *BeaconR_Ack* au nœud émetteur correspondant et passe automatiquement à l'état *WAIT_DATA*; sinon, il retourne à l'état *SLEEP*. Si aucun *DATA_packet* n'est reçu du nœud émetteur correspondant pendant *WAIT_DATA*, le nœud passe immédiatement à l'état *SLEEP*; sinon, il retourne à l'état *SEND_BEACON_ACK*; sinon, si le nœud reçoit *Beacon* ou *Beacon_Ack* ou *Beacon_R* ou *BeaconR_Ack*, il reste dans l'état actuel. Une fois que le destinataire envoie une balise *Beacon_Ack* au nœud émetteur, il passe automatiquement à l'état *SLEEP*.

5.6.2 Les paramètres de simulation

Le *RCSF* est construit avec *MiXiM* à l'aide de la topologie de grille afin d'éviter les surcoûts probables et les différents paramètres sont spécifiés dans le fichier *omnetpp.ini*. Ces paramètres, indiqués dans les tableaux 5.2 et 5.3, incluent le nombre total de nœuds, le nœud de destination, le débit de trafic, paramètre de trafic (Traffic Param) (il s'agit de l'intervalle de temps maximal entre deux transmissions de paquets exprimé en secondes), le type de réseau, le type d'application, la taille du playground, les paramètres de la batterie (nominal, capacité et nombre de batteries) et les paramètres de récupération d'énergie qui sont: le type de récolte, le paramètre de la récolte (il s'agit de l'intervalle de temps maximal entre deux moments de récolte d'énergie exprimé en secondes) et le taux de la récolte. Les paramètres de la couche physique, y compris les modèles analogiques et le type de décideur, sont définis dans le fichier *config.xml* et le type *MAC* est déclaré dans le fichier *Nic.ned*. De nombreux scénarios sont utilisés et mis en œuvre, allant de 16 nœuds déployés dans une zone de 400 m X 400 m à 100 nœuds de capteur déployés dans une zone de 1100 m X 1100 m dans lesquels la distance entre deux nœuds voisins dans la grille est de 100 m pour 1000 s de temps simulé où tous les nœuds de capteurs source peuvent envoyer leur trafic périodique toutes les 1,5 ou 2 s vers le récepteur par le biais de transmissions multi-sauts. Le taux de récupération d'énergie est périodique (toutes les 1,5 s) où la récolte d'énergie de chaque nœud de capteur est défini de manière aléatoire entre [0 mW...50 mW], car la plupart des sources d'énergie ambiante ont des

caractéristiques diverses qui dépendent des différentes situations environnementales. L'énergie initiale de chaque nœud de capteur est fixée à 50 J et la valeur maximale de l'énergie est de 100 J , la valeur du premier seuil d'énergie E_{th_1} est réglée à 35 J et la valeur du deuxième seuil d'énergie E_{th_2} réglée à 30 J .

Couche	Paramètre	Valeur
<i>Générale</i>	Nombre de nœuds	16, 25, 49, 64, 100
	Topologie	Grille
	Temps de simulation	1000 s
	Taille du playground (x,y)	(400 m , 400 m), (600 m , 600 m), (800 m , 800 m), (950 m , 950 m), (1100 m , 1100 m)
	Energie résiduelle initiale E_{rem_i}	50 J
	Seuil d'énergie E_{th_1}	35 J
	Seuil d'énergie E_{th_2}	30 J
	MaxAttempts	2
<i>Application</i>	Type d'application	Sensor App Layer
	Type de Trafic	Périodique
	Paramètre de trafic (Traffic Param)	1.5 s , 2.0 s (secondes)
	Nombre de paquets envoyés	1000
<i>Réseau</i>	Type de réseau	BaseNetwLayer
	Nœud de destination	15, 24, 48, 63, 99
<i>MAC</i>	Type de MAC	<i>EEMEHWSNLayer</i> <i>DSRLayer</i> , <i>DSPLayer</i> <i>RIMACLAYER</i>
	Taille de la file d'attente	10 <i>pkts</i> (packets)
<i>Physique</i>	Puissance de transmission	0.05 W
	Fréquence de la porteuse	2.412e+9 HZ
	Modèle de propagation	SimplePathlossModel

Tableau 5.2: Les paramètres de simulation à différentes couches pour l'implémentation du *EEM-EHWSN*

Energie	Paramètre	Valeur
<i>Puissance (Batterie)</i>	Nominale de la batterie	100 J
	Capacité de la batterie	100 J
	Nombre de batterie	1
<i>Récolte</i>	Type la récolte énergétique	Périodique
	Paramètre de la récolte	1.5 s
	Taux de la récolte aléatoire	[0 mW ...50 mW]

Tableau 5.3: Les paramètres de simulation de la puissance énergétique et de la récolte d'énergie pour l'implémentation du *EEM-EHWSN*

5.7 Évaluation des performances

Par le biais de simulations approfondies utilisant le simulateur *OMNeT++* avec le modèle de simulation *MiXiM*, nous avons comparé notre nouveau schéma de gestion de l'énergie (*EEM-EHWSN*) qui utilise un schéma de planification de duty-cycle dynamique tout d'abord par rapport à un schéma de planification de duty-cycle statique appelé *RIMAC* qui est la base de tous les protocoles de récupération d'énergie du côté de la communication entre les nœuds, car il s'agit d'un protocole d'efficacité énergétique et de montrer également que le duty-cycle dynamique utilisé par notre schéma (*EEM-EHWSN*) est plus avantageux que le duty-cycle statique utilisé par *RIMAC* dans le but d'améliorer les performances du *RCSF* et leur durée de vie. Deuxièmement, par rapport à deux schémas de planification de duty-cycle dynamiques, appelés *DSR* et *DSP*, utilisant un duty-cycle dynamique, comme *EEM-EHWSN*, mais ils ne garantissent pas l'équilibrage de la charge dans le réseau pour lequel l'intervalle de réveil maximal utilisé par les nœuds de capteur I_{dc}^{max} est défini à 1 s. D'autre part, notre schéma assure l'équilibrage de la charge en calculant la période de veille et par l'utilisation de la technique de commutation proposée utilisée par chaque nœud dans la période de réveil. Les métriques de performance utilisées pour évaluer les protocoles sont les suivantes : le taux de livraison des paquets qui devrait être maximisé, la latence moyenne des paquets qui devrait être minimisée et le débit qui devrait être maximisé.

Nous rapportons les résultats des expériences de simulation pour le protocole *RIMAC*, les schémas *DSP* et *DSR* et notre schéma *EEM-EHWSN* comme suit:

La figure 5.10 représente les résultats de la simulation pour la métrique de taux de livraison de paquets avec des valeurs de paramètres de trafic (1,5 et 2,0 s) et avec un nombre de paquets envoyés allant jusqu'à 1000 alors que la taille du réseau est comprise entre 16 et 100 nœuds. Les résultats indiquent que le taux de livraison des paquets diminue avec l'augmentation du nombre de nœuds. De plus, les résultats montrent que *EEM-EHWSN* est en mesure de produire un taux de livraison de paquets plus élevé que les deux schémas *DSP* et *DSR*, qui utilisent le duty-cycle dynamique pour différents taux de la récolte énergétique grâce à l'ajustement dynamique des duty-cycles en fonction de l'énergie restante, qui peut augmenter avec le temps. Dans *EEM-EHWSN*, les périodes de veille sont choisies en connaissance de cause, ce qui est peu probable pour être la même pour un voisinage. De plus, dans *DSP* et *DSR*, la latence de sommeil est réduite. Cela réduit efficacement la collision et la retransmission de paquets. Ceci est dû au fait que *EEM-EHWSN* favorise davantage l'équilibrage de charge dans *RCSF* que *DSP* et *DSR* en déterminant la période de veille appropriée de chaque nœud en fonction de son énergie restante et de l'utilisation de la technique de commutation, limitant ainsi davantage la collision. D'autre part, *RIMAC* fournit un faible taux de livraison de paquets dans les mêmes scénarios, en raison de l'utilisation du duty-cycle statique. Par exemple, dans le cas où le paramètre de trafic est défini sur 1,5 s et que le nombre de nœuds augmente de 16 à 49, comme indiqué sur la figure 5.10(a), *EEM-EHWSN* surpasse *DSP* et *DSR* en termes de taux de livraison de paquets compris entre 18,43 % et 11,21 %, 18,73 % et

35,85 % respectivement. Pour *EEMEHWSN*, la figure 5.10(a) montre un gain en termes de taux de livraison de paquets supérieur (environ 28,11 % et 58,88 % par comparaison avec *DSP* et *DSR* respectivement) lorsque le paramètre de trafic = 1,5 s et le nombre de nœuds = 100. De même, la figure 5.10(b) montre un gain en termes de taux de livraison plus élevé d'environ 14,44 % et 44,46 % respectivement lorsque le paramètre de trafic = 2,0 s et le nombre de nœuds = 100. On peut donc constater que des gains de performance significatifs en termes de taux de livraison de paquets sont obtenus avec *EEMEHWSN* plus qu'avec *DSR* et *DSP*.

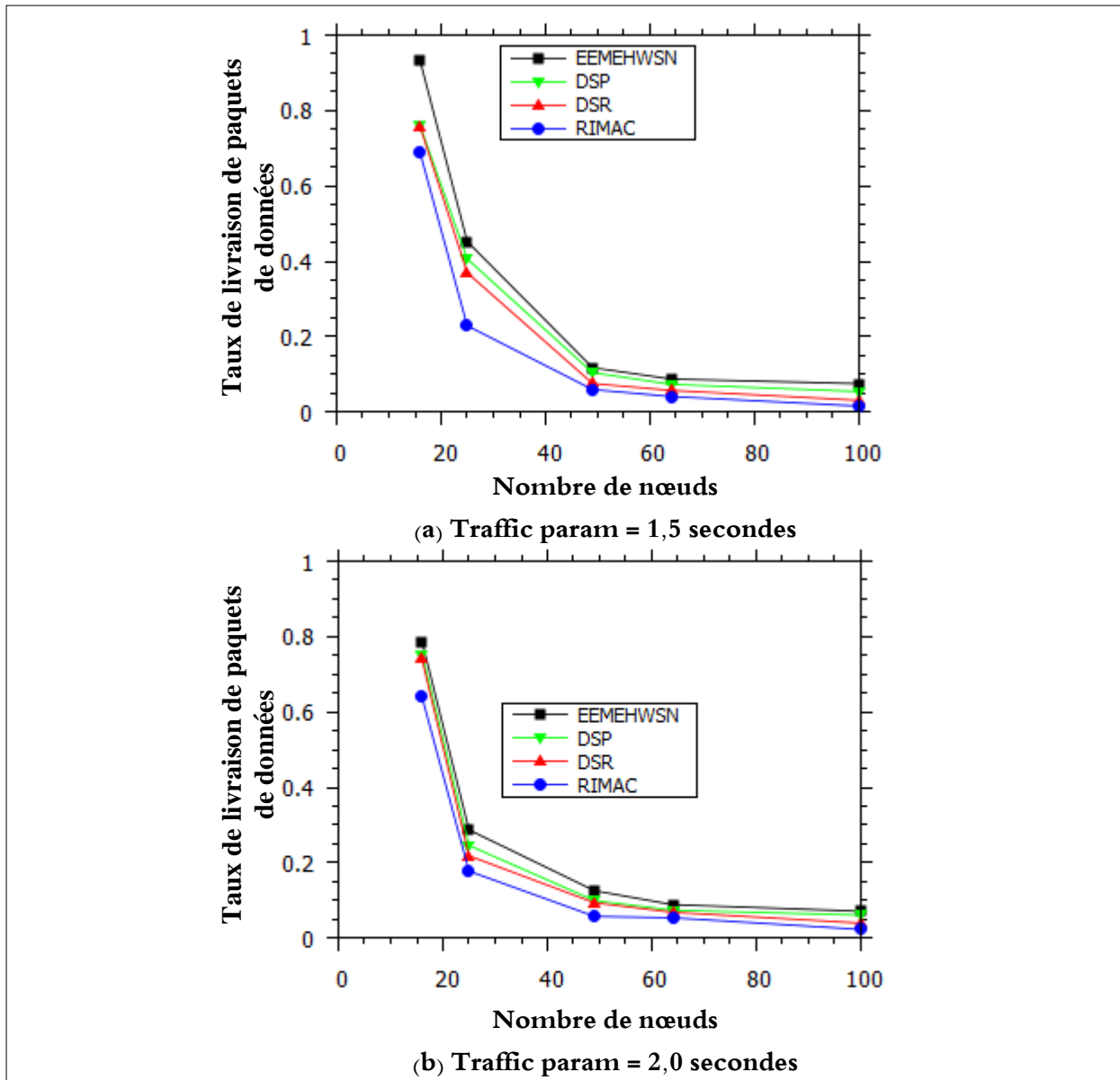


Figure 5.10: Le taux de livraison des paquets. (a) paramètre de trafic = 1.5 s, (b) paramètre de trafic = 2.0 s

La figure 5.11 présente les résultats pour la métrique de latence moyenne des paquets. Nous observons que la latence moyenne augmente lorsque le nombre de nœuds augmente. Les résultats montrent que *RIMAC* introduit une latence moyenne plus élevée que *EEMEHWSN*, *DSP* et *DSR*, en raison de l'utilisation du duty-cycle statique. Inversement, la latence moyenne du *EEMEHWSN* est inférieure à celle de *DSP* et de

DSR. Les deux raisons sont que notre schéma encourage les nœuds ayant un niveau d'énergie élevé à être actifs en raison de leur durée de période de veille réduite, calculée sur la base de l'équation (5.14) et l'utilisation de la technique de commutation afin d'assurer un équilibrage de charge dans le RCSF, réduisant ainsi le nombre de collisions. Par conséquent, il est moins probable que les paquets de données soient mis en file d'attente, ce qui réduit le temps de latence moyenne par rapport à DSP et DSR. De plus, seule la latence de sommeil est réduite pour DSR et DSP, mais aucun équilibrage de la charge n'est pris en compte. Pour EEMEHWSN, la figure 5.11(a) montre un gain en termes de latence moyenne inférieure d'environ 38,63 % et 41,01 % lorsque le paramètre de trafic = 1,5 s et le nombre de nœuds = 100 par comparaison avec DSP et DSR, respectivement. De la même manière, la figure 5.11(b) montre un gain en termes de latence moyenne inférieure d'environ 48,66 % et 54,19 % lorsque le paramètre de trafic = 2,0 s et le nombre de nœuds = 100.

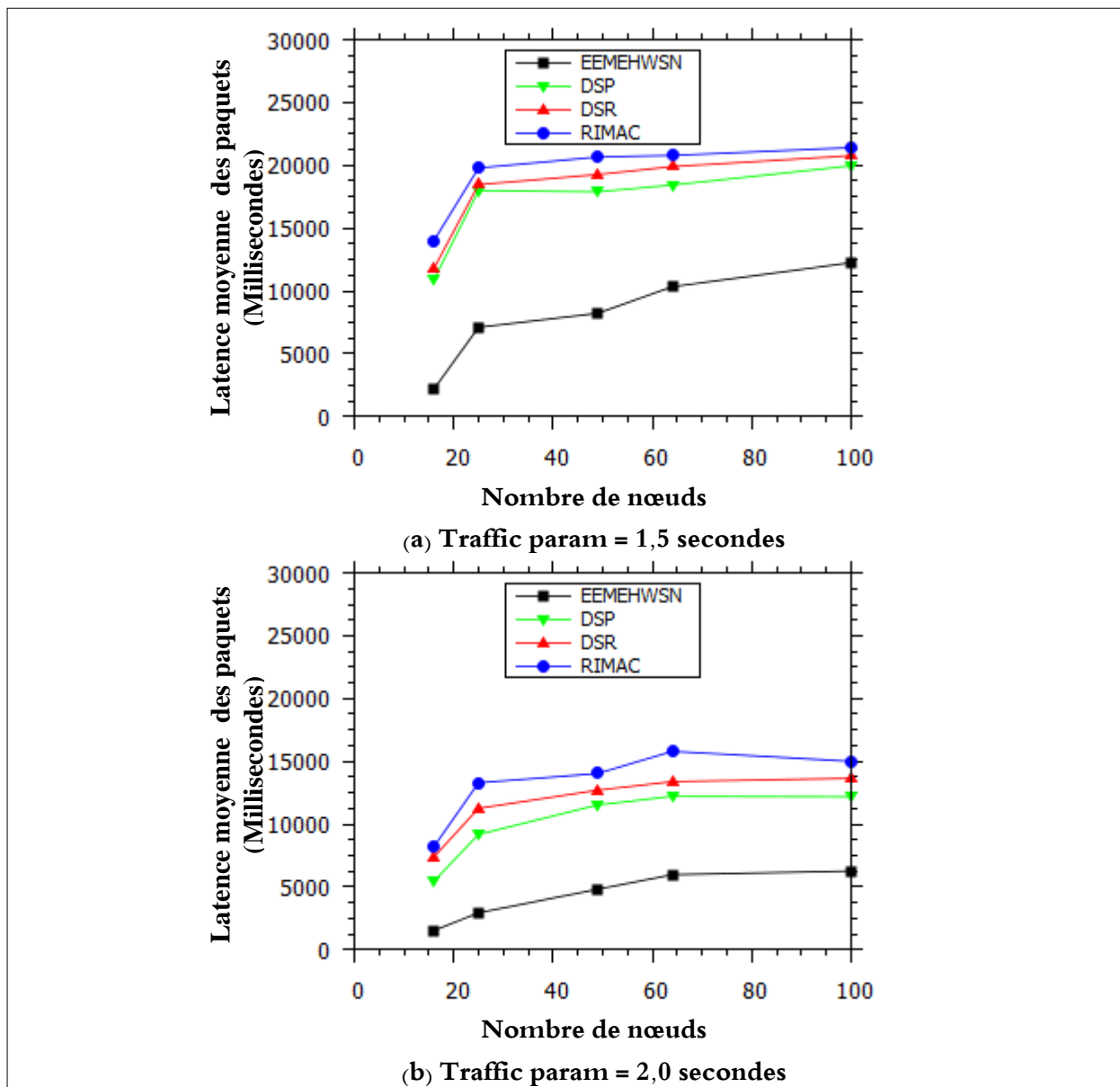


Figure 5.11: La latence moyenne des paquets. (a) paramètre de trafic = 1.5 s, (b) paramètre de trafic = 2.0 s

La figure 5.12 représente les valeurs de débit calculées en bits par seconde (*bps*) générées par un nombre variable de nœuds. Le résultat indique que le débit diminue avec l'augmentation du nombre de nœuds. De plus, les résultats montrent que *EEMEHWSN* est en mesure de fournir un débit supérieur à celui des autres deux schémas *DSP* et *DSR* grâce à la sélection dynamique des temps de veille des nœuds de capteurs qui sont très peu susceptibles d'être égaux, ce qui dépend de l'ajustement dynamique des duty-cycles sur la base de la quantité d'énergie récupérée et en plus de l'utilisation de la technique de commutation proposée. Ceci assure un équilibre de charge dans le *RCSF*; réduisant ainsi davantage la collision et le nombre de retransmissions que *DSP* et *DSR*. Parce que, plus il y a de collisions, plus le temps est nécessaire pour une transmission réussie.

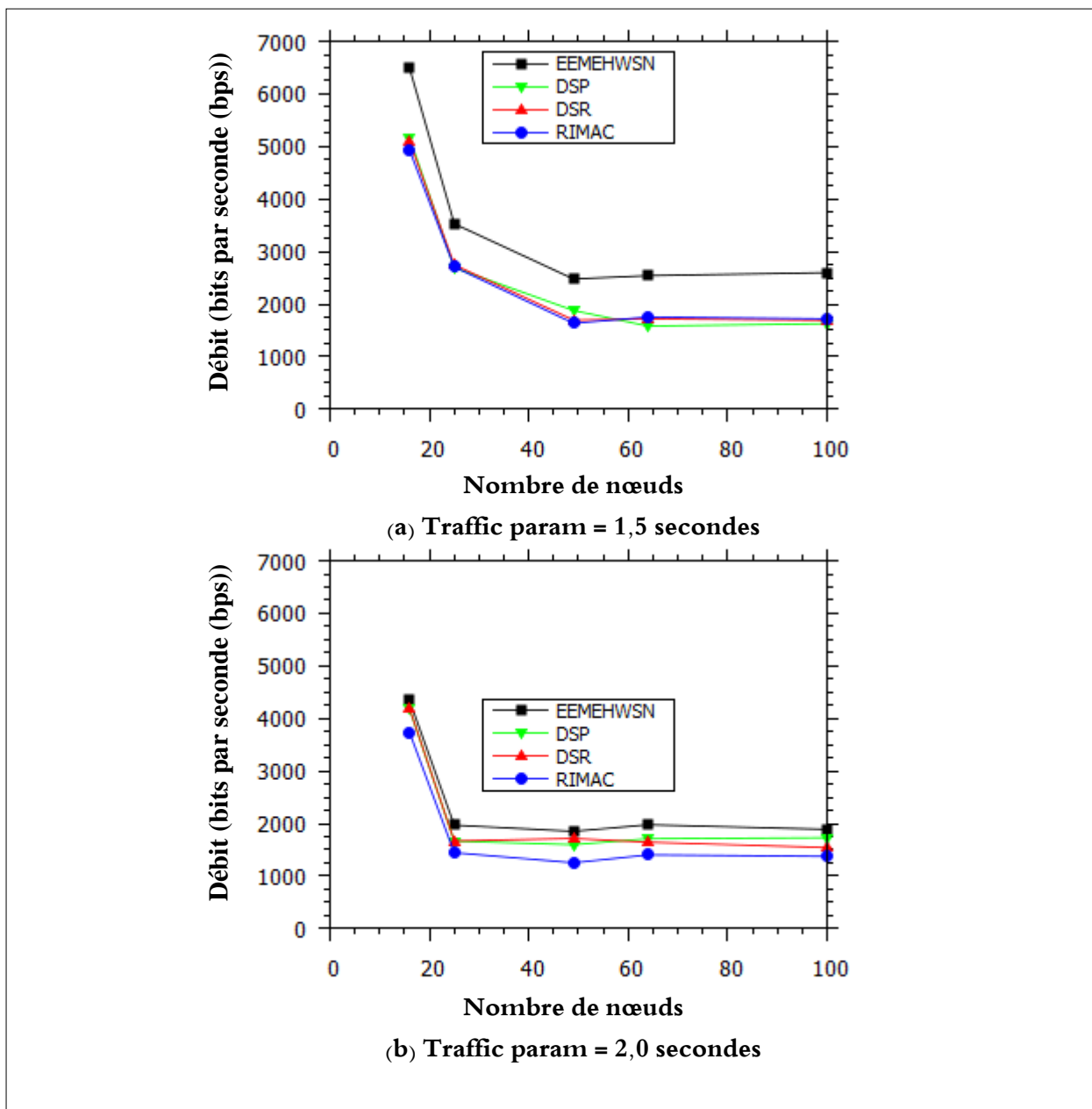


Figure 5.12: Le débit. (a) paramètre de trafic = 1.5 s, (b) paramètre de trafic = 2.0 s

Par exemple, tout d'abord sur la figure 5.12(a), on peut remarquer que des gains de performances significatifs (environ 37,42 % et 34,96 %) du débit ont été obtenus par *EEMEHWSN* par comparaison avec *DSP* et *DSR* respectivement, lorsque le nombre de nœuds = 100 et avec le paramètre de trafic égale à 1,5 s. Deuxièmement, sur la figure 5.12(b), on peut noter que des gains de performances significatifs (environ 8,38 % et 18,30 %) sont comparés avec *DSP* et *DSR* respectivement lorsque le nombre de nœuds = 100 et avec le paramètre de trafic réglé à 2,0 s. D'autre part, *RIMAC* fournit un faible débit dans les mêmes conditions que pour *EEMEHWSN*, *DSP* et *DSR*, en raison de l'utilisation du duty-cycle statique.

Par conséquent, afin d'augmenter les performances d'un *RCSF*, en termes de taux de livraison de paquets, de latence moyenne de paquets et de débit moyen dans le domaine de la récupération énergétique, la réduction du duty-cycle et l'équilibrage de la charge sont importants pour éviter la consommation d'énergie provoquée par la collision et la période d'activité élevée d'un nœud.

Dans l'ensemble, en observant les résultats des scénarios mentionnés ci-dessus, nous pouvons conclure que les performances du schéma *RIMAC* sont faibles. Par la suite, le schéma *DSP* est efficace car il réduit le duty-cycle de manière plus agressive que *DSR* sur la base de son énergie restante, où la latence du sommeil est réduite sans prendre en compte l'équilibrage de la charge, mais pas mieux que notre nouveau schéma *EEMEHWSN*. Avec *EEMEHWSN*, le taux de livraison moyen de paquets est plus élevé et la latence moyenne de paquets est inférieure à celui des systèmes *DSP* et *DSR* car il utilise le duty-cycle dynamique dont la durée de la période de veille est calculée et utilise également la technique de commutation proposée, assurant ainsi l'équilibrage de la charge dans le *RCSF*. Nous pouvons donc en conclure que *EEMEHWSN* présente des résultats prometteurs par rapport aux deux autres schémas *DSP* et *DSR*.

5.8 Conclusion

Pour les réseaux de capteurs sans fil, l'alimentation est un élément crucial. Il y a essentiellement deux aspects: premièrement, le stockage d'énergie et la livraison de l'énergie sous la forme requise; deuxièmement, la tentative de récupération de l'énergie consommée par le réapprovisionnement via une source externe vers le nœud du capteur, telle que des cellules solaires. Dans ce chapitre, nous avons proposé un système de gestion de l'énergie amélioré pour les réseaux de capteurs sans fil à récupération d'énergie (*EEM-EHWSN*).

Ce travail comporte deux contributions principales basées sur l'utilisation de la politique de seuil d'énergie utilisant deux seuils E_{th_1} et E_{th_2} . La première utilise E_{th_1} , est la proposition d'une technique de commutation permettant à chaque nœud de capteur de basculer entre le mode de détection et le mode de relais pendant l'état de réveil basé sur une comparaison entre E_{th_1} et l'énergie restante au niveau de la batterie du nœud pouvant être augmentée au fil du temps afin d'assurer un équilibrage de la charge dans le *RCSF*. La seconde utilise E_{th_2} , offrant un duty-cycle dynamique basé sur l'utilisation du principe

de notre premier mécanisme *EH2M* dans laquelle la durée de la période de veille est calculée sur la base de l'interpolation de Lagrange (équation (5.14)) tant que l'énergie restante est supérieure ou égale à E_{th_2} afin de réduire le temps de latence moyen et augmente le débit et le taux de livraison de paquets dans le *RCSF*.

Afin de valider les performances de notre schéma *EEM-EHWSN* proposé, nous avons implémenté notre schéma *EEMEHWSN*, les schémas *DSP*, *DSR* et le protocole *RIMAC* à l'aide du simulateur *OMNeT++* et le modèle de simulation *MiXiM*. Les trois premiers schémas (*EEMEHWSN*, *DSP* et *DSR*) sont les premiers schémas de récupération d'énergie implémentés dans le simulateur *OMNeT++ / MiXiM*. Via la simulation dans *OMNeT++ / MiXiM*, le *EEMEHWSN* est évalué avec de nombreux scénarios de tests et avec des flux de trafic variables. Dans toutes les expériences, *EEMEHWSN* surpasse de manière significative les autres schémas (les deux schémas *DSP* et *DSR* et le protocole *RIMAC*) en termes de taux de livraison des paquets, de latence moyenne de paquets et de débit, qui sont les mesures de performance du *RCSF* les plus importantes. Simultanément, *EEMEHWSN* assure un équilibre entre la consommation d'énergie et la capacité de récupération d'énergie de chaque nœud de capteur dans le *RCSF*. Le chapitre suivant décrit un autre nouveau protocole *MAC* de gestion de l'énergie proposé pour les *RCSF-RE* qui est basé sur le cycle d'activité (duty-cycle) dynamique d'un nœud capteur dans le réseau sur la base de sa quantité d'énergie restante dont la communication est initiée par l'émetteur contrairement au *EEM-EHWSN* dans le but d'améliorer les performances du *RCSF*. Ainsi, l'amélioration de sa durée de vie.

PS-EHWSN : Un protocole MAC basé sur un schéma d'échantillonnage de préambule pour les RCSF-RE

6.1 Introduction

Le fonctionnement des *RCSF* avec des ressources énergétiques limitées est un défi important pour les chercheurs. En fait, de nombreux projets ont été proposés pour économiser l'énergie gaspillée dans un réseau ou pour équiper ses nœuds de capteurs d'un système de récupération d'énergie.

Dans le chapitre précédent nous avons présenté notre nouveau schéma de gestion de l'énergie *EEM-EHWSN* proposé pour les *RCSF-RE* qui est basé sur la planification du duty-cycle dynamique et sur la technique de commutation entre deux modes (de détection et relais) utilisées par les nœuds afin de réduire les duty-cycles des nœuds dans le réseau et d'assurer un équilibrage de charge entre eux dont la communication est initiée par le récepteur. Le souci majeur du *EEM-EHWSN* est qu'il ne peut pas assurer la communication dans les *RCSF-RE* dans le cas où il n'y a pas suffisamment de nœuds de transfert éveillés, soit parce que la densité des nœuds en service est trop basse ou encore que la période de récolte moyenne d'énergie est trop longue. De ce fait, notre objectif dans ce chapitre est d'exposer un autre nouveau protocole dédiée pour les *RCSF-RE* mobile et qui représente un prélude à l'invention d'un autre protocole du type *DTN* (*Delay Tolerant Network*) [126].

Dans ce chapitre, un protocole *MAC* basé sur un schéma d'échantillonnage de préambule pour les *RCSF-RE* nommé *PS-EHWSN* (*A Preamble Sampling Scheme MAC Protocol for Energy Harvesting WSNs*) [124] est présenté. *PS-EHWSN* utilise la méthode du préambule strobé en diffusant une séquence de paquets de préambule courts. Des petites pauses entre les paquets de préambule permettent au voisinage du destinataire d'envoyer un accusé de réception qui arrête la séquence de paquets de préambule. Il fonctionne selon un schéma de planification de duty-cycle dynamique, qui exploite l'énergie récupérée par les nœuds.

6.2 Architecture du nœud de capteur utilisée

Comme dans notre schéma *EEM-EHWSN* décrit dans le chapitre précédent, notre nouveau protocole *PS-EHWSN* utilise une architecture du nœud de capteur similaire à celle utilisée par *EEM-EHWSN* présentée dans la section 5.2 du chapitre 5 où la différence réside dans le plan de gestion de mobilité qui va être exploité car cette dernière est destinée pour le type de *RCSF* mobile.

6.3 Le schéma de communication fondamental utilisé dans PS-EHWSN

Similaire que *EEM-EHWSN*, le *PS-EHWSN* utilise un protocole *CSMA/CA* dans lequel l'expéditeur utilise l'onde porteuse pour rendre le canal occupé et le destinataire écoute le canal sans fil pour éviter les collisions. Dans le *PS-EHWSN*, tous les nœuds qui ont des paquets en file d'attente devant être transférés au récepteur, écoutent le canal en attente d'un accusé de réception (*ACK*) approprié après chaque diffusion du préambule court. Lors de la réception de l'*ACK* correspondant, la transmission du paquet de données est lancée.

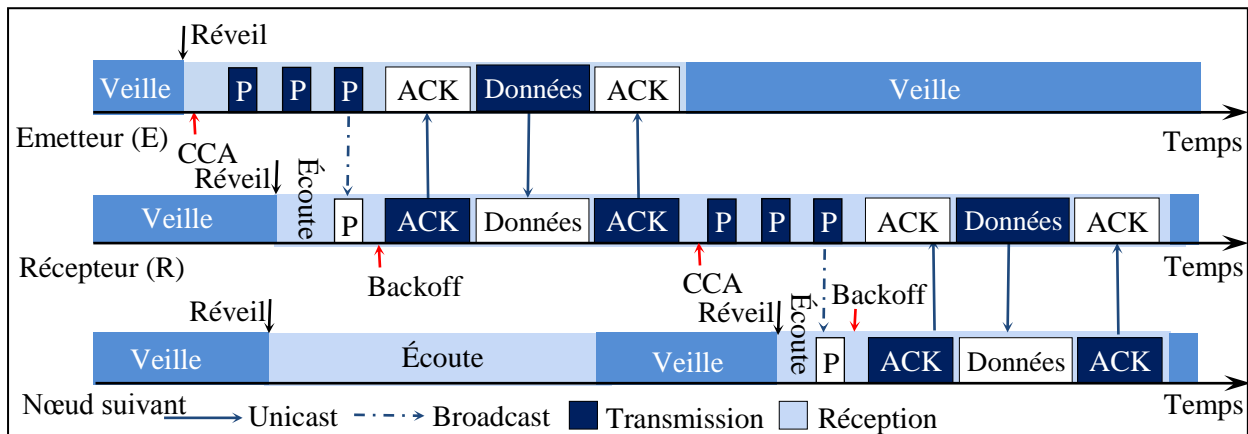


Figure 6.1: Illustration du fonctionnement de notre protocole *PS-EHWSN*

Notre protocole proposé est fondé aussi sur le duty-cycle, qui associe les périodes d'activité et de veille, offre aux récupérateurs d'énergie une opportunité respectable de recharger la source d'énergie avant de basculer à un état de déchargement où l'énergie déposée est consommée par le nœud du capteur. Le *PS-EHWSN* repose sur le principe de communication suivant, illustré à la figure 6.1:

1. les nœuds de notre protocole se réveillent périodiquement pour effectuer l'opération de transmission s'il y a des données à envoyer et basculer en état de veille lorsqu'il n'y a pas de trafic sur le réseau après un temps d'écoute ou après chaque transmission de données successive;
2. tous les nœuds basculent vers l'état de veille au démarrage, puis sur l'état d'écoute en fonction de leur propre programme pour écouter le trafic sur le réseau;
3. si un nœud émetteur *E* souhaite transmettre des données, il initie une session de préambule en envoyant en diffusion plusieurs paquets de préambule court *P* aux nœuds voisins après avoir vérifié la disponibilité du support (medium) à l'aide du *CCA* (*Clear Channel Assessment*) et se met en pause après chaque transmission de préambule court pour écouter le canal. Un nœud récepteur *R* réveillé répond à l'émetteur en envoyant un paquet *ACK* (*accusé de réception du préambule*) en mode monodiffusion après avoir reçu un préambule court si leur *back off* est expiré;
4. l'expéditeur *E* cesse d'envoyer les préambules courts restants *P* après la réception de l'accusé de réception *ACK* et commence à envoyer le paquet de données au destinataire prévu si le canal est libre, ce qui sera acquitté par *R* avec *ACK*; sinon, il

passé en état de veille après une durée prédéfinie allouée à l'envoi de plusieurs préambules courts P ;

5. l'émetteur E bascule en mode veille après la réception de l'accusé de réception du paquet de données ACK ;

Ensuite, le nœud récepteur commence immédiatement à transmettre ses données au nœud suivant en exécutant les étapes 3, 4 et 5.

6.4 Le modèle de la consommation d'énergie et de la latence

Symbole	Signification
E_{cs}	Énergie de détection de la porteuse
E_{es}	Énergie de détection de l'environnement
E_{rx}	Énergie de réception
E_{tx}	Énergie de transmission
P_{cs}	Puissance de détection de la porteuse
P_{es}	Puissance de détection de l'environnement
P_{rx}	Puissance de réception
P_{tx}	Puissance de transmission
R_{data}	Taux de RX/TX d'un paquet de données
R_{es}	Taux pour détecter l'environnement
L_{packet}	Longueur du paquet de données
$L_{spreamble}$	Longueur du paquet préambule court
L_{ack}	Longueur de l'accusé de réception des données (ACK)
T_{cs}	Temps de détection de porteuse
T_{es}	Temps de détection de l'environnement
T_{acs}	Temps moyen de détection de la porteuse
T_{asense}	Temps moyen pour échantillonner les données
T_{rx}	Temps de réception
T_{tx}	Temps de transmission
$T_{(t/r)xb}$	Temps de Tx/Rx par octet
T_{sp}	Temps pour l'envoi d'un préambule court
T_{sw}	Temps de commutation radio
T_{que}	Temps d'attente
T_{pro}	Temps de traitement

Tableau 6.1: Les termes utilisés pour les modèles : réseau, l'énergie et la latence dans PS-EHWSN

Nous supposons que nous avons un réseau de N nœuds mobiles et que nous considérons un type de trafic périodique engendrant, où les nœuds produisent des données avec un intervalle périodique constant. Toutes les notations des termes utilisés pour les modèles de réseau, d'énergie et de latence qui seront utilisés dans cette section sont mentionnées dans le tableau 6.1.

6.4.1 Le modèle de la consommation d'énergie

La consommation d'énergie anticipée dans l'état de réveil pendant l'intervalle t par un nœud i dans *PS-EHWSN*, est la somme de la consommation d'énergie anticipée dans chaque état: détection de la porteuse, détection d'environnement, émission et réception.

$$\begin{aligned} E_i(t) &= E_{cs}(t) + E_{es}(t) + E_{tx}(t) + E_{rx}(t) \\ &= T_{cs}P_{cs} + T_{es}P_{es} + T_{tx}P_{tx} + T_{rx}P_{rx} \end{aligned} \quad (6.1)$$

À chaque période de réveil, un nœud détecte la porteuse avant d'envoyer un préambule. Il détecte également l'environnement pour surveiller les valeurs physiques.

$$T_{cs} = T_{acs}R_{data} \quad (6.2)$$

$$T_{es} = T_{asense}R_{es} \quad (6.3)$$

Le temps de transmission du nœud i est la somme du temps nécessaire pour envoyer des paquets de données, des préambules courts et des accusés de réception (d'un préambule court et d'un paquet de données). Chaque fois qu'il a des données à envoyer, il envoie un court préambule dans un temps T_{sp} requis et change immédiatement sa radio en mode écoute afin de recevoir l'accusé de réception *ACK*. Ce processus est répété jusqu'à ce qu'il reçoive un accusé de réception du nœud cible. K est le nombre de tentatives qu'un nœud envoie un préambule court et change sa radio pour recevoir un accusé de réception *ACK*. Après réception du paquet de données, le récepteur change de sa radio pour envoyer un *ACK*.

$$L_{spreamble} = T_{SP}/T_{(t/r)xb} \quad (6.4)$$

$$T_{tx} = R_{data}(L_{packet} + 2L_{ack})T_{(t/r)xb} + KT_{sp} + (K + 2)T_{sw} \quad (6.5)$$

Un nœud peut recevoir plusieurs préambules (Σ), paquets de données et accusés de réception au cours d'une période donnée. Mais lorsqu'il devient un nœud cible pour un préambule spécifique, il envoie immédiatement un accusé de réception à l'expéditeur et reçoit son paquet, puis envoie un accusé de réception de ce dernier.

$$T_{rx} = R_{data}(L_{packet} + 2L_{ack})T_{(t/r)xb} + \Sigma T_{sp} + 2T_{sw} \quad (6.6)$$

6.4.2 Le modèle de la latence

Après la mise en file d'attente et le traitement de chaque paquet de données dans chaque nœud. Le nœud reçoit un accusé de réception après l'envoi de plusieurs préambules courts avant l'envoi de chaque paquet de données et envoie un accusé de réception après la réception de chaque paquet de données. Selon les équations. (6.2), (6.5) et (6.6), la latence attendue est:

$$\begin{aligned}
 L &= T_{que} + T_{pro} + T_{cs} + T_{tx} + T_{rx} \\
 &= T_{que} + T_{pro} + T_{cs} + 2R_{data}(L_{packet} + 2L_{ack})t_{(t/r)xb} + (K + \Sigma)T_{sp} + \\
 &\quad (K + 4)T_{SW}
 \end{aligned} \tag{6.7}$$

6.5 Les fonctions de base du PS-EHWSN

Comme dans notre schéma *EEM-EHWSN* décrit dans le chapitre précédent, notre nouveau protocole *PS-EHWSN* utilise aussi une politique de seuil d'énergie basée seulement sur le mécanisme de gestion de l'énergie (*EH2M*) présenté dans la section 5.5.1 du chapitre 5 pour calculer la durée de la période de veille d'un nœud dans le réseau sur la base de leur énergie résiduelle. *PS-EHWSN* fonctionne selon deux modes, le mode statique où les durées des deux périodes (veille et réveil) sont déterminées par l'administrateur et le mode dynamique utilisé dans le reste de leur fonctionnement dans lequel le duty-cycle de chaque nœud dans le réseau est ajusté selon la quantité d'énergie récupérée et de leur énergie résiduelle. *PS-EHWSN* favorise les nœuds ayant une quantité d'énergie résiduelle plus grande pour participer dans la communication dont leur durée de la période de veille est plus courte. Dans *PS-EHWSN* la durée de la période de veille d'un nœud est choisie (calculée) consciemment en fonction de l'énergie résiduelle, il minimise ainsi, la collision dans le voisinage produit par le réveil des nœuds en même temps et se concurrent le canal. En conséquence, un nœud possédant une énergie résiduelle faible choisit un intervalle de veille long pour récupérer encore mieux d'énergie. *PS-EHWSN* est basé sur les fonctions de base suivantes, comme indiqué sur la figure 6.2:

1. initialement, chaque nœud de capteur i bascule en mode veille au démarrage et après chaque instant de vérification t_{check} , il calcule son énergie restante $E_{remi}(t_{check})$ à l'aide de l'équation (5.10) dont $E_{consumed_i}(t_{check} - t'_{check})$ est calculée en utilisant l'équation (6.1);
2. si l'énergie restante $E_{remi}(t_{check})$ est inférieure au seuil d'énergie E_{th} , il initialise son t_{is_i} (l'instant de démarrage de la récupération d'énergie); sinon, si t_{is_i} est initialisé, il calcule d'abord son t_{csi} et ensuite son d_{th_i} (la durée de la période de veille écoulée jusqu'à ce que l'énergie restante atteigne le seuil énergétique E_{th}) grâce à l'utilisation de l'équation (5.12) sinon, il vérifie directement la condition pour passer en état de réveil;
3. il passe en état de réveil selon l'une des deux conditions de commutation suivantes:
 - 1- si l'énergie restante $E_{remi}(t_{check})$ est supérieure ou égale au seuil d'énergie E_{th} après avoir récolté la quantité d'énergie requise où le d_{th_i} est calculée.
 - 2- si la durée de la période de veille d_{sleep_i} est expirée définie par l'administrateur dans le mode de duty-cycle statique qui ne sera utilisée qu'une seule fois par chaque nœud capteur pendant le fonctionnement du réseau tant que l'énergie restante est supérieure ou égale au seuil d'énergie E_{th} , ou il est calculé périodiquement dans le mode de duty-cycle dynamique sur la base de l'équation (5.14) et l'équation (6.1) en fonction de la quantité d'énergie récupérée et de l'énergie restante $E_{remi}(t_{check})$ lorsque cette dernière est supérieure ou égale au seuil d'énergie E_{th} avant qu'il soit en état de veille, qui sera utilisé dans le reste de fonctionnement du *PS-EHWSN*;

4. si la condition de basculement à l'état de veille est vérifiée après l'utilisation du système de communication (réception ou transmission de (*Preamble*, *Ackk* (Accusé de réception de préambule), *Ack*, ou paquet de données)) appliqué par le protocole PS-EHWSN et utilisé par chaque nœud de capteur i , ce dernier calcule la nouvelle prochaine durée de la période de veille d_{sleep_i} grâce à l'utilisation de l'équation (5.14) sur la base de l'énergie restante $E_{remi}(t_{check})$ qui est calculée à l'aide de l'équation (5.10) dont $E_{consumed_i}(t_{check} - t'_{check})$ est calculée en utilisant l'équation (6.1) si la valeur restante de l'énergie $E_{remi}(t_{check})$ est supérieure ou égale au seuil d'énergie E_{th} ou il utilise la durée de la période de veille définie par l'administrateur pour le mode de duty-cycle statique ou calculée dans le mode dynamique précédent si d_{th_i} (la durée de la période de veille écoulée jusqu'à ce que l'énergie restante $E_{remi}(t_{check})$ atteigne le seuil d'énergie E_{th}) n'est pas calculée.

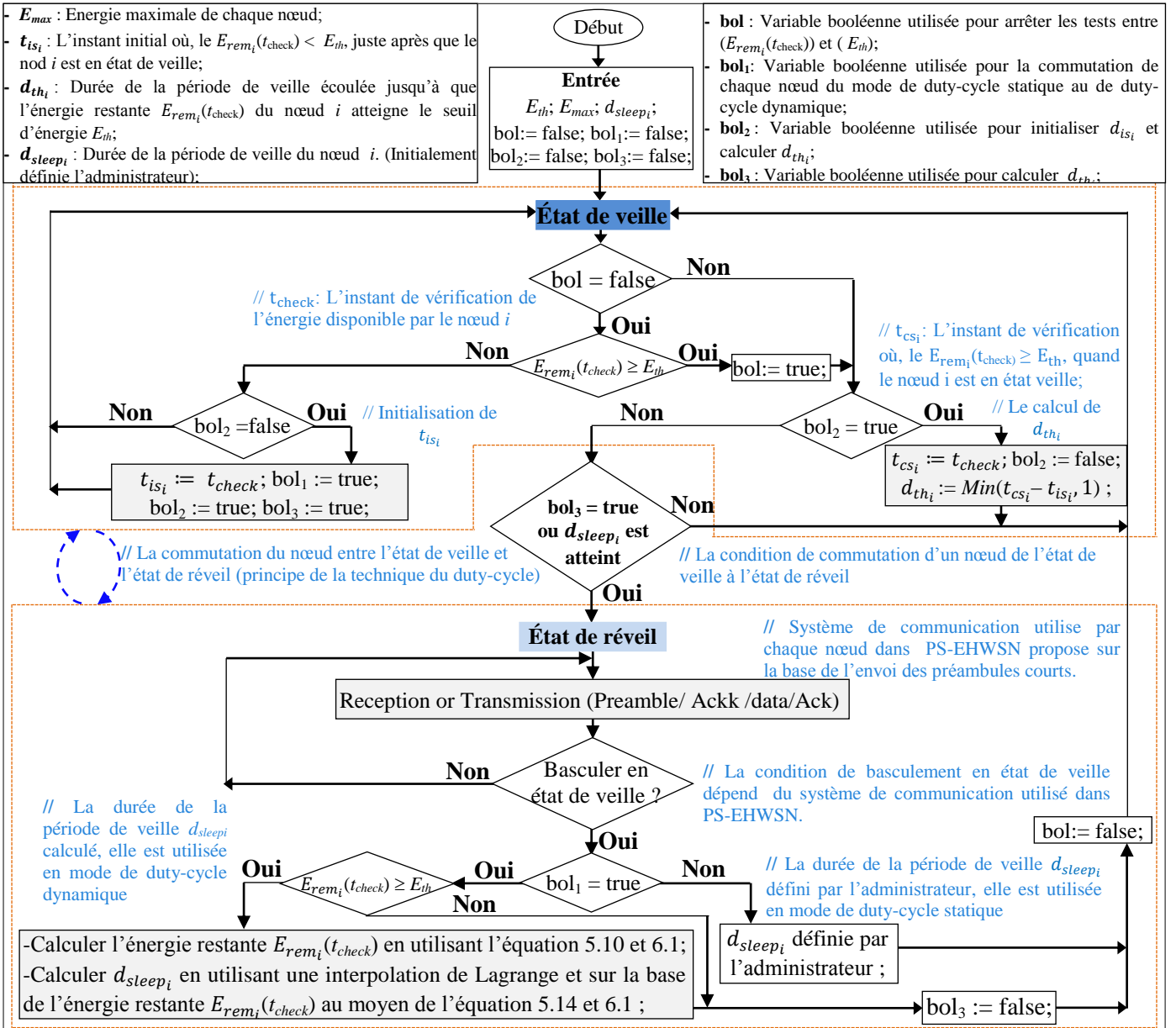


Figure 6.2: Fonctions de base du protocole PS-EHWSN où chaque nœud de capteur est équipé d'un système de récupération d'énergie.

6.6 La simulation

Comme dans les deux chapitres précédents, nous avons utilisé pour la mise en œuvre du protocole proposé *PS-EHWSN* la plate-forme de simulation *OMNeT ++-4.6* avec le cadre de simulation *MiXiM-2.3*. La principale implémentation de *PS-EHWSN* est *PSEHWSNLayer*, un module de la couche *MAC* intégré dans une nouvelle interface réseau *NicPSEHWSN* (basée sur *WirelessNicBattery* (voir la section 4.4 du chapitre 4), composée d'un ensemble de modules spécifiques tels que *BatteryStats* qui calcule la consommation d'énergie de chaque module, de l'interface réseau *WirelessNicBattery* et du bloc d'alimentation *SimpleBattery*).

Un nouvel hôte *HostPSEHWSN* (basé sur *WirelessNodeBattery* et sur le module de la récolte énergétique) contenant cette interface réseau est définie à la figure 6.3. La création d'un réseau *HostPSEHWSN* et le réglage des paramètres réseau ce sont deux moyens utiles pour évaluer les performances de *PS-EHWSN*. *PSEHWSNLayer* contient des portes permettant de se connecter à d'autres modules ou d'autres couches. Les messages sont utilisés non seulement en tant que paquets à envoyer entre les couches d'un nœud de capteur ou entre deux nœuds, mais également en tant qu'événements permettant de modifier l'état d'un nœud.

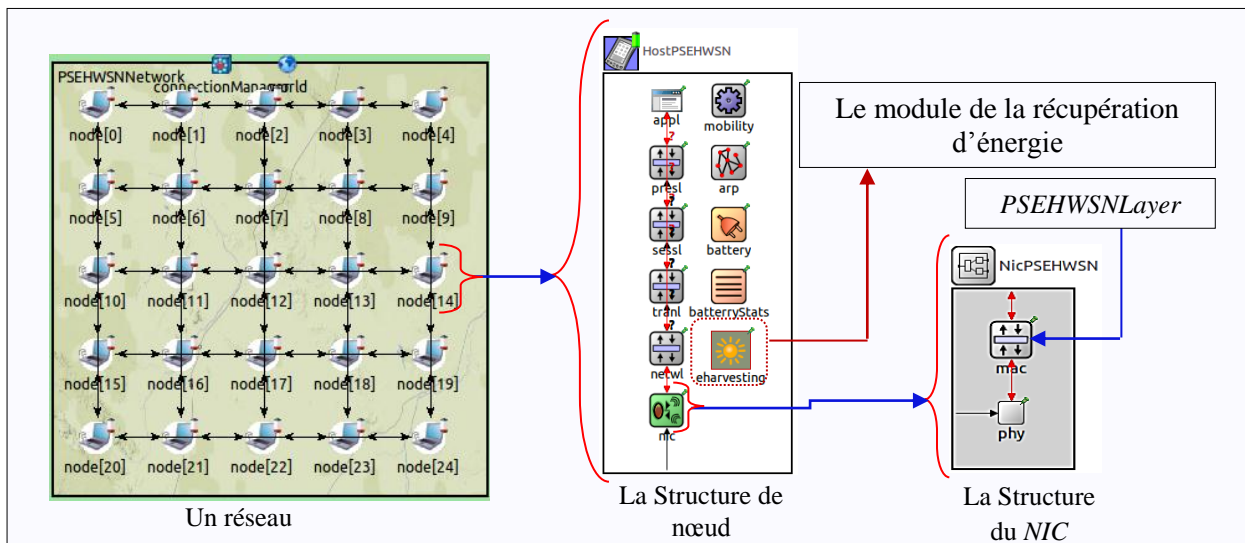


Figure 6.3: Le réseau du protocole PS-EHWSN

6.6.1 La machine à états finis du nœud émetteur et récepteur avec *PS-EHWSN*

Nous présentons dans cette section la machine à états finis du nœud émetteur et récepteur représenté à la figure 6.4, qui illustre tous les messages: auto-messages (*selfmessages*) et les paquets réseau utilisés dans *PSEHWSNLayer* pour modifier l'état d'un nœud *HostPSEHWSN*. Après initialisation dans l'état *INIT*, le nœud passe à l'état *SLEEP*. Tout d'abord, nous expliquons la machine à états finis du point de l'émetteur.

Pour le nœud émetteur. Ce dernier ayant un paquet à envoyer basculera d'abord dans l'état *CCA* à partir de l'état *SLEEP* et initialise la valeur de la variable « *permit* » à

true. Si aucune communication n'est détectée pendant CCA (*CCA_timeout* est atteint), il passe à l'état *SEND_PREAMBLE* et commence à envoyer des paquets de préambule court. Après chaque transmission de préambule, il passe à l'état *WAIT_ACKK* si le délai d'envoi du préambule n'est pas atteint (*StopPreambles*); sinon, il retourne à l'état *SLEEP*. Le nœud passera à l'état *SEND_DATA* après avoir reçu un accusé de réception *ACKK* du préambule. Une fois que l'expéditeur envoie le paquet de données *DATA* au nœud récepteur (si le canal est libre), il passe automatiquement à l'état *WAIT_ACK* immédiatement. Enfin, pendant l'état *WAIT_ACK* si le nœud reçoit (un accusé de réception de données) ou (*ACK_timeout* est atteint et *TxAttempts* > *MaxAttempts*), il bascule automatiquement à l'état *SLEEP* immédiatement; sinon, si (*ACK_timeout* est atteint et que *TxAttempts* < *MaxAttempts*), il retourne à l'état *SEND_PREAMBLE*.

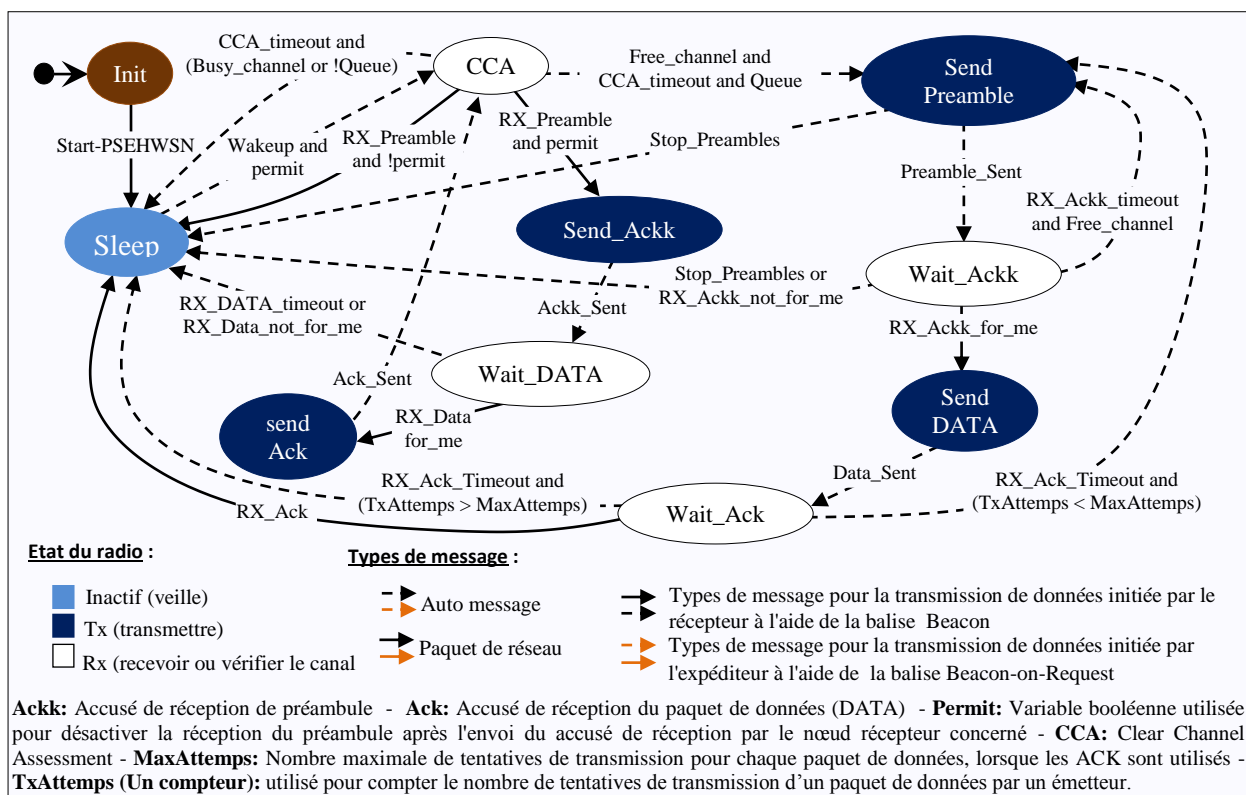


Figure 6.4: Machine à états finis de HostPSEHWSN

Du côté du nœud récepteur, il se réveille pour détecter le canal en vue d'une écoute périodique en passant de l'état *SLEEP* à l'état *CCA*. Si le nœud reçoit un préambule de ses voisins et que la valeur de la variable « *permit* » est égale à *true* lors de l'état *CCA*, il passe à l'état *SEND_ACKK* pour communiquer avec le nœud émetteur correspondant lorsque leur *back off* est expiré et passe immédiatement à l'état *WAIT_DATA*; sinon, il retourne à l'état *SLEEP*. Si aucun paquet de données n'est reçu du nœud émetteur correspondant pendant *WAIT_DATA*, le nœud passe immédiatement à l'état *SLEEP*; sinon, il retourne à l'état *SEND_ACK* et change la valeur de la variable « *permit* » à *false*. Une fois que le destinataire a envoyé un accusé de réception au nœud émetteur, il passe automatiquement à l'état *CCA* pour envoyer immédiatement le paquet de données reçu au nœud suivant.

6.6.2 Les paramètres de simulation utilisés

Couche	Paramètre	Valeur
<i>Générale</i>	Nombre de nœuds	10, 25,50, 75, 100
	Taille du playground (x,y)	(400 m, 400 m), (600 m, 600 m), (800 m, 800 m), (950 m, 950 m), (1100 m, 1100 m)
	Type de mobilité	ConstSpeedMobility
	Vitesse de mobilité	2 mps
	Energie résiduelle initiale E_{rem_i}	50 J
	Seuil d'énergie E_{th}	30 J
	MaxAttempts	2
<i>Application</i>	Type d'application	Sensor App Layer
	Type de Trafic	Périodique
	Paramètre de trafic (Traffic Param)	0.1 s, 0.5 s
	Nombre de paquets envoyés	500, 1000
<i>Réseau</i>	Type de réseau	BaseNetwLayer
	Nœud de destination	9, 24, 49, 74, 99
<i>MAC</i>	Type de MAC	<i>PSEHWSNLayer</i> <i>BMACLayer</i>
	Taille de la file d'attente	10 pkts (packets)
<i>Physique</i>	Puissance de transmission	0.05 W
	Fréquence de la porteuse	2.412e+9 HZ
	Modèle de propagation	SimplePathlossModel

Tableau 6.2: Les paramètres de simulation à différentes couches pour l'implémentation du PS-EHWSN

Energie	Paramètre	Valeur
<i>Puissance (Batterie)</i>	Nominale de la batterie	100 J
	Capacité de la batterie	100 J
	Nombre de batterie	1
<i>Récolte</i>	Type la récolte énergétique	Périodique
	Paramètre de la récolte	0.1 s
	Taux de la récolte aléatoire	[0 mW...50 mW]

Tableau 6.3: Les paramètres de simulation de la puissance énergétique et de la récolte d'énergie pour l'implémentation du PS-EHWSN

Comme *EEM-EHWSN* et avec la plate-forme de simulation *OMNeT ++-4.6* et le cadre de simulation *MiXiM-2.3*, un *RCSF* est construit pour évaluer notre protocole *PS-EHWSN*. Les différents paramètres utilisés dans la simulation sont résumés dans les deux tableaux 6.2 et 6.3, spécifiés dans le fichier *omnetpp.ini*. De plus, les paramètres de la couche physique, y compris les modèles analogiques et le type de décision, sont déclarés dans le fichier *config.xml* et le type de schéma *MAC* est spécifié dans le fichier *Nic.ned*. Ici le Traffic Param est l'intervalle de temps maximum entre la transmission de deux

paquets exprimé en secondes et Harvesting Param (paramètre de la récolte énergétique) est l'intervalle de temps maximum entre deux moments de récolte d'énergie également exprimé en secondes. De nombreux scénarios sont mis en œuvre et testés, allant de 10 nœuds de capteurs mobiles déployés dans une zone de $400\text{ m} \times 400\text{ m}$ à 100 nœuds de capteurs mobiles déployés dans une zone de $1100\text{ m} \times 1100\text{ m}$ où la distance entre deux nœuds voisins est de 100 mètres pendant 500 secondes simulées. Le taux de récupération d'énergie est périodique avec Harvesting Param est égal à 0,1 seconde pour chaque nœud de capteur réglé de manière aléatoire entre $[0\text{ mW} \dots 50\text{ mW}]$. L'énergie initiale de chaque nœud de capteur mobile est définie sur 50 J et la valeur maximale de l'énergie est de 100 J , la valeur du seuil d'énergie E_{th} est réglée à 30 J .

6.7 Évaluation des performances

Grâce à des simulations approfondies utilisant *OMNeT++* avec *MiXiM*, nous avons comparé notre protocole *PSEHWSN* à celui de *BMAC* qui est le premier protocole fondé sur un schéma d'échantillonnage de préambule et basé aussi sur la diffusion comme *PS-EHWSN* mais il utilise dans cette diffusion un long préambule aux nœuds voisins dans le réseau. Les métriques de performance utilisées pour évaluer les deux protocoles sont les suivantes: le taux de livraison des paquets qui devrait être maximisé, la latence moyenne des paquets qui devrait être minimisée et le débit qui devrait être maximisé.

Nous remettons les résultats des expériences de simulation pour le protocole *BMAC* et notre protocole *PSEHWSN* comme suit:

La figure 6.5 représente les résultats de la simulation pour la métrique de taux de livraison de paquets avec des diverses valeurs de paramètre de trafic (TrafficParam) ($0,1\text{ s}$ et $0,5\text{ s}$) selon un nombre différent de paquets envoyés (500 et 1000) où le nombre de nœuds est compris entre 10 et 100. Les résultats indiquent que le taux de livraison de paquets diminue avec l'augmentation du nombre de nœuds. De plus, les résultats montrent que *PSEHWSN* est en mesure de fournir un taux de livraison de paquets élevé par rapport à *BMAC* pour différents taux de récupération d'énergie, en ajustant les duty-cycles de manière dynamique en fonction de l'énergie résiduelle qui peut augmenter avec le temps. Cela est également dû à l'utilisation de la méthode de l'envoi des préambules courts. D'autre part, *BMAC* fournit un taux de livraison de paquets faible avec l'augmentation de la charge de trafic, avec les mêmes paramètres que ceux utilisés par *PSEHWSN*, en raison de l'utilisation du duty-cycle statique; de plus, la transmission d'une longue séquence de préambule continue avant la transmission du paquet de données nuit davantage à ses performances et provoque la chute de paquets due à une collision. Par exemple, dans le cas où le nombre de paquets envoyés est défini sur 500 et que le nombre de nœuds augmente de 10 à 50, comme indiqué à la figure 6.5(a), *PSEHWSN* surpasse *BMAC* en termes de taux de livraison des paquets compris environ entre 95,74 % et 97,36 % lorsque TrafficParam est défini sur $0,1\text{ s}$ et entre 84,53 % et 89,12 % lorsque TrafficParam est défini sur $0,5\text{ s}$. De plus, le taux de livraison des paquets compris entre environ 93 % et 95,13 % lorsque TrafficParam est défini sur $0,1\text{ s}$ et entre 56,87% et

78,24 % lorsque TrafficParam est défini sur 0,5 s dans le cas où le nombre de paquets envoyés est défini sur 1000 comme le montre la figure 6.5(b). Pour *PSEHWSN*, la figure 6.5(a) montre un gain en termes de taux de livraison élevé d'environ 97,28 % lorsque TrafficParam = 0,1 s et 87,85 % lorsque TrafficParam = 0,5 s par comparaison avec *BMAC* lorsque le nombre de nœuds est égal à 100 et le nombre de paquets envoyés = 500. De même, la figure 6.5(b) montre un gain en termes de taux de livraison élevé d'environ 95,89 % lorsque TrafficParam = 0,1 s et 75,70 % lorsque TrafficParam = 0,5 s lorsque la taille du réseau est définie sur 100 et que le nombre des paquets envoyés = 1000. Par conséquent, on peut constater que *PSEHWSN* par comparaison à *BMAC* permet d'obtenir des gains de performance significatifs en termes de taux de livraison.

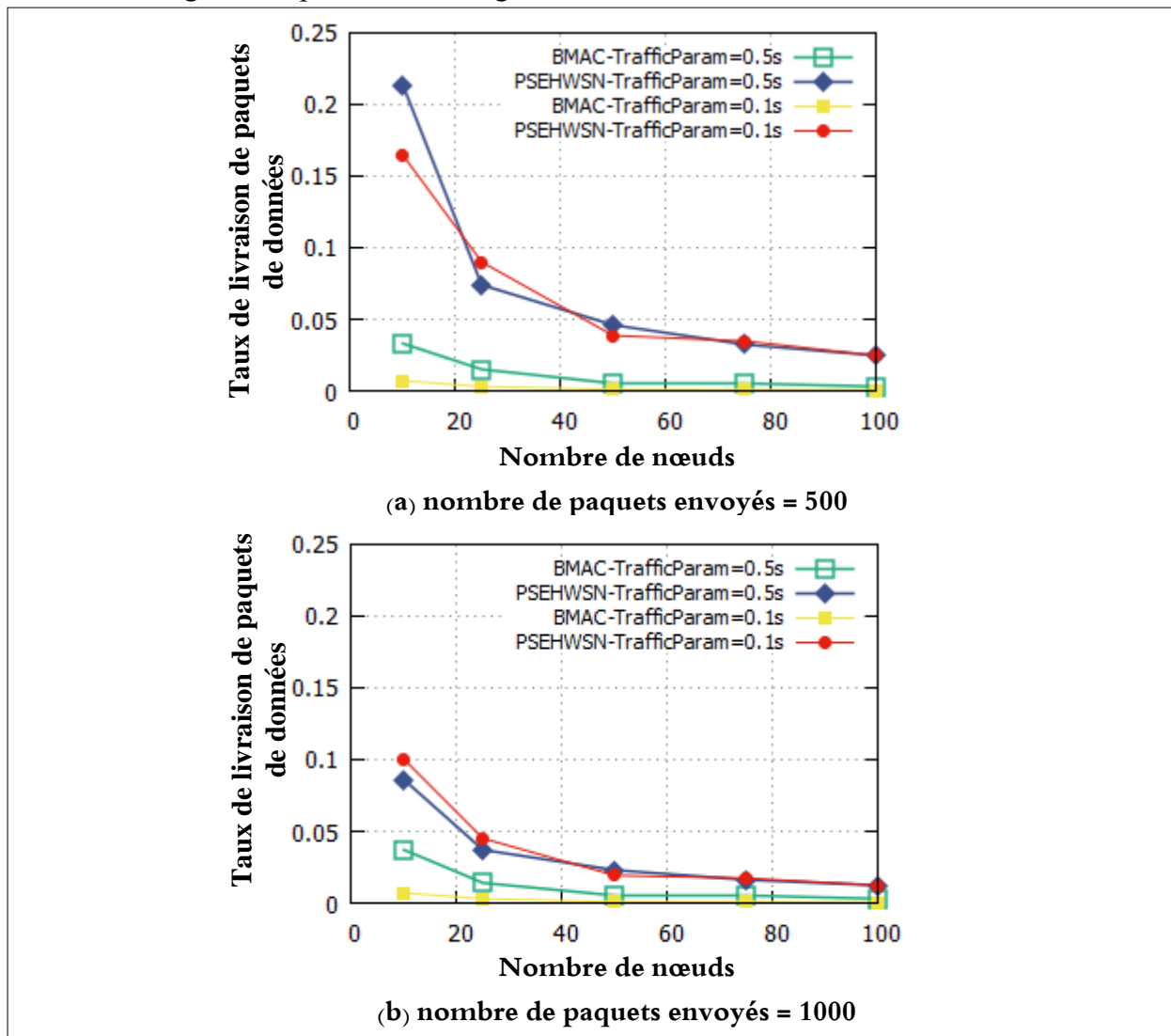


Figure 6.5: Le taux de livraison des paquets. (a) Nombre de paquets envoyés = 500, (b) Nombre de paquets envoyés = 1000

La figure 6.6 présente les résultats obtenus pour la métrique de latence moyenne. Nous observons que la latence moyenne augmente lorsque le nombre de nœuds augmente. Les résultats montrent que *PSEHWSN* surpasse légèrement *BMAC* lorsque le nombre de nœuds augmente. De plus, on peut percevoir que la latence moyenne de *PSEHWSN* est inférieure à celle de *BMAC*. Les deux raisons sont que notre protocole

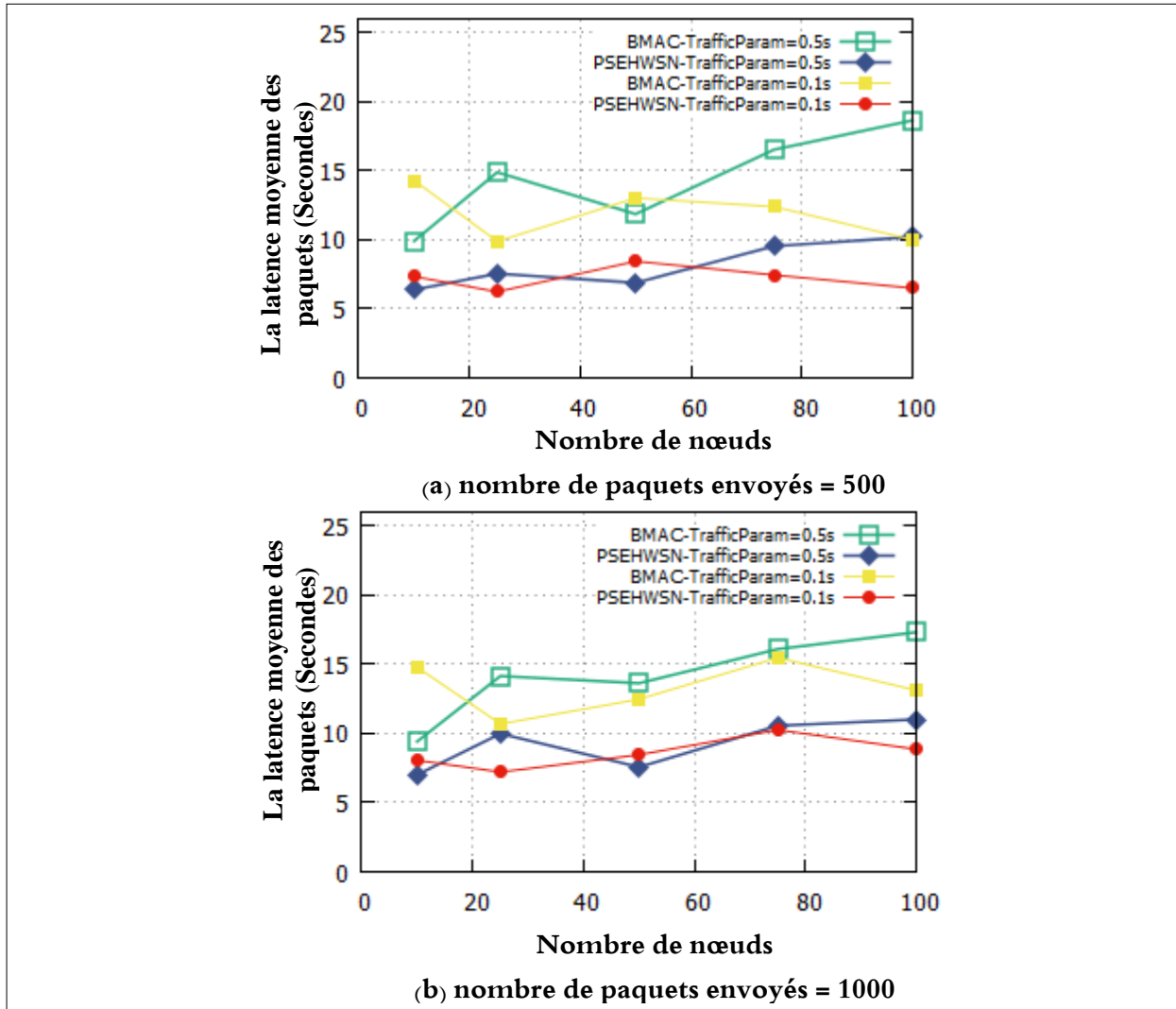


Figure 6.6: La latence moyenne des paquets. (a) Nombre de paquets envoyés = 500, (b) Nombre de paquets envoyés = 1000

PSEHWSN encourage les nœuds ayant un niveau d'énergie élevé à être actifs en raison de leur durée de période de veille réduite, calculée sur la base de l'équation (5.14) dont $E_{consumed_i}(t_{check} - t'_{check})$ dans l'équation (5.10) est calculée en utilisant l'équation (6.1) et de l'utilisation de la méthode de l'envoi des préambules courts. Cela produit plus de chances que les paquets de données ne soient pas mis en file d'attente et réduit considérablement le temps de latence moyen par rapport à la transmission d'une longue séquence de préambule continue utilisée par *BMAC*. D'autre part, notre mécanisme proposé utilise une méthodologie d'approximation de la durée de la période de veille des nœuds pour ajuster leur duty-cycle sur la base de l'énergie récoltée présentée à l'avenir. Pour *PSEHWSN*, la figure 6.6(a) montre un gain en termes de latence moyenne inférieure d'environ 34,93 % lorsque TrafficParam = 0,1 s et 45,27 % lorsque TrafficParam = 0,5 s par rapport à *BMAC* lorsque le nombre de nœuds est égal à 100 et le nombre de paquets envoyés = 500. De même, la figure 6.6(b) montre un gain en termes de temps de latence moyen inférieur d'environ 32,49 % lorsque TrafficParam = 0,1 s et 36,39 % lorsque TrafficParam = 0,5 s lorsque la taille du réseau est définie sur 100 et que le nombre des paquets envoyés égale à 1000.

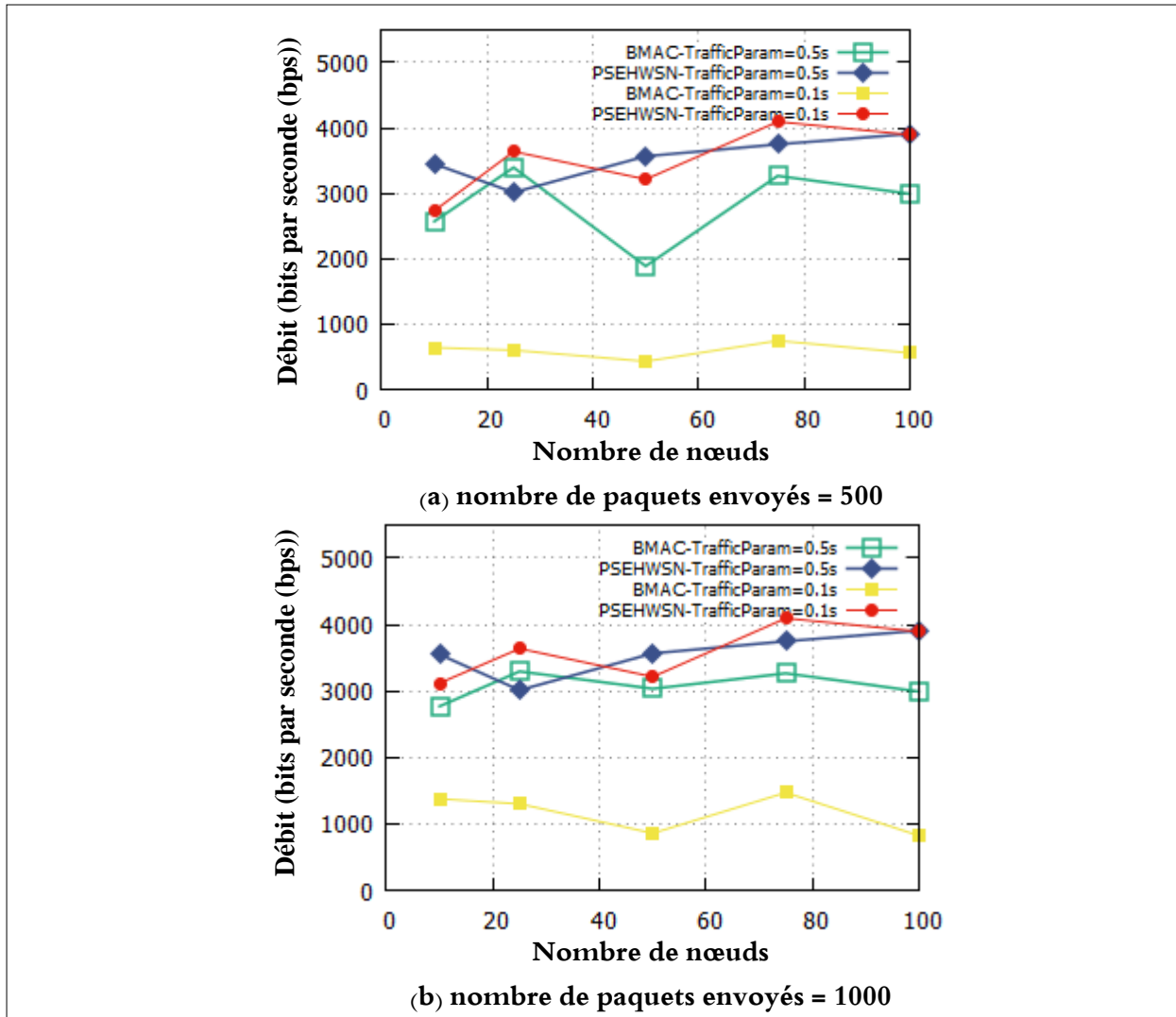


Figure 6.7: Le débit. (a) Nombre de paquets envoyés = 500, (b) Nombre de paquets envoyés = 1000

La figure 6.7 représente la valeur de débit mesurée en bits par seconde (*bps*) générée par un nombre variable de nœuds. Le résultat indique que le débit augmente avec l'augmentation du nombre de nœuds. De plus, les résultats montrent que *PSEHWSN* est en mesure de fournir un débit supérieur à *BMAC* grâce à la diminution dynamique de la période de réveil des nœuds de capteurs, qui dépend de la quantité d'énergie récupérée et permet l'ajustement dynamique des duty-cycles et en raison de l'utilisation de la méthode de l'envoi des préambules courts. Cela réduit la collision et le nombre maximal de retransmissions, car plus il y a de collisions, plus il faut de temps pour qu'une transmission réussie qui entraîne la perte de paquets. D'autre part, le *BMAC* fournit un faible débit de paquets en raison de la transmission d'une longue séquence de préambule continue avant la transmission du paquet de données, ce qui provoque une chute de paquet due à une collision. Par exemple, tout d'abord sur la figure 6.7(a), lorsque le nombre de nœuds augmente de 50 à 100, le débit augmente de 17,46 % pour *PSEHWSN* et de 22,98 % pour *BMAC*, pour lequel TrafficParam est défini sur 0,1 s et le nombre de paquets envoyés défini sur 500, on peut remarquer sur cette figure que des gains de performances significatifs d'environ 85,69 % du débit ont été obtenus de *PSEHWSN* par

rapport à *BMAC*, lorsque le nombre de nœuds est égal à 100 et que TrafficParam est défini sur 0,5 s. Deuxièmement, sur la figure 6.7(b), lorsque le nombre de nœuds augmente de 50 à 100, le débit augmente de 17,46 % pour *PSEHWSN* et de 5 % pour *BMAC*, dont TrafficParam est défini sur 0,1 s et le nombre de paquets envoyés = 1000. De plus, on remarque de cette figure que des gains de performances significatifs d'environ 78,88 % du débit ont été obtenus avec *PSEHWSN* par rapport à *BMAC*, où le nombre de nœuds = 100. De même lorsque le TrafficParam est défini sur 0,5 s.

En conséquence, grâce à la réduction de duty-cycle variable qui le rend dynamique, *PS-EHWSN* dépasse le *BMAC* qui utilise un duty-cycle statique et une transmission de paquets de préambule plus long que les périodes de veille des nœuds récepteurs causant une chute des paquets due à la collision, ce qui fait que les nœuds récepteurs restent éveillés jusqu'à la fin de ce long préambule. Globalement, en observant les résultats des scénarios mentionnés ci-dessus, on peut constater que les gains de performances significatifs en termes de débit, de latence moyenne et de taux de livraison des paquets sont générés par *PS-EHWSN* par comparaison à *BMAC*.

6.8 Conclusion

Les *RCSF* mobiles se caractérisent par leur absence d'infrastructure, leur dynamique et leurs ressources limitées telles que la puissance de traitement; le stockage et surtout la source d'énergie car ils sont généralement alimentés par des batteries. Recharger les batteries des nœuds de capteurs est parfois impossible en raison de leurs emplacements inaccessibles. Ce chapitre décrit le protocole *PS-EHWSN* adapté aux *RCSF* mobiles et qui constitue une nouvelle approche de la communication à faible puissance dans ce type de réseau basée sur la récolte énergétique. *PS-EHWSN* utilise l'approche basée sur la diffusion d'une série de paquets de préambule courts et fonctionne selon un schéma de planification dynamique du duty-cycle comme dans le schéma *EEM-EHWSN*. Notre protocole *PS-EHWSN* définit la durée de la prochaine période de veille d'un nœud i par l'utilisation du niveau d'énergie résiduelle. Le calcul de cette durée est basé sur l'interpolation de Lagrange (équation (5.14)) et sur l'énergie résiduelle $E_{remi}(t_{check})$ calculée à l'aide de l'équation (5.10) dont $E_{consumed_i}(t_{check} - t'_{check})$ est calculée en utilisant l'équation (6.1); chaque nœud i pouvant ajuster son duty-cycle de manière dynamique. Ce concept améliore les performances du réseau par rapport au protocole *BMAC* de différentes manières. En comparant les performances de *PS-EHWSN* à celles du protocole *BMAC* via des simulations avec *OMNeT++/MiXiM* avec de nombreux de test et avec des flux de trafic variables, *PS-EHWSN* est capable de gérer l'énergie récupérée d'un nœud de capteur utilisé pour la détection, la transmission, la réception et le traitement; il réduit le temps de latence moyen, à un taux de livraison des paquets plus élevé et un débit plus élevé car le duty-cycle de chaque nœud du réseau dépend de la valeur du taux de récupération d'énergie.

Conclusion générale et perspectives

1 Conclusion générale

Les *RCSF* jouent un rôle essentiel dans de nombreuses applications, telles que l'automatisation de processus industriels, les systèmes de contrôle aérien, le contrôle des patients dans des hôpitaux et la surveillance de l'environnement. Au fil des années, il a été prouvé que la gestion de l'énergie dans ce type de réseau prolongeait la durée de vie des capteurs grâce à l'utilisation des schémas de conservation d'énergie innovants, tel que le schéma de « *duty-cycling* » qui a pour objet de basculer un nœud de capteur dans le réseau en mode veille en éteignant sa radio de communication le plus possible afin de réduire la consommation d'énergie, si ce dernier n'est pas inclus dans la communication courante sinon il le bascule en mode réveil, mais la quantité d'énergie requise par les capteurs pour fonctionner pendant longtemps demeure un défi. Il est donc essentiel d'utiliser des récupérateurs d'énergie pour soutenir les systèmes de gestion de l'énergie existants. Dans les travaux effectués dans cette thèse de doctorat, on a étudié le problème de gestion de l'énergie issu d'un *RCSF* dans lequel les nœuds de capteurs sont dotés d'un système de récupération d'énergie qui leur permettait de récupérer de l'énergie à partir des sources d'énergie ambiantes de l'environnement externe (lumière solaire et artificielle, mouvements vibratoires, chaleur,...etc.) et de convertir cette énergie en énergie électrique emmagasinée afin de servir à leur fonctionnement dans le but d'améliorer la durée de vie du *RCSF*. L'objectif de ce travail de thèse consiste principalement à adapter le fonctionnement du réseau aux cycles de recharge décharge des capteurs par la proposition d'un système de veille-réveil adapté à la fréquence de recharge de chaque capteur ainsi que de l'importance du nœud capteur dans l'acheminement des paquets.

En premier lieu, nous avons commencé notre thèse par donner une introduction générale dans le domaine des *RCSF* dans laquelle la motivation et la problématique sont présentées. Dans le chapitre 1, nous avons donné un aperçu sur les *RCSF* où nous avons décrit les différentes couches de la pile protocolaire utilisée par un nœud de capteur et leurs différentes sources de consommation d'énergie dans ce type de réseau et par la suite certaines définitions sont données telles que la couverture, la connectivité et la durée de vie du *RCSF*. Dans le chapitre 2, nous avons commencé une étude complète de plusieurs protocoles *MAC* pour les *RCSF* fonctionnant sous différents schémas d'accès au canal et nous les avons analysés pour plusieurs métriques, ce qui montre que la plupart des protocoles s'appliquent à l'efficacité énergétique. Ensuite, nous avons fait une autre étude complète de certains protocoles *MAC* pour les *RCSF-RE* dans le chapitre 3 où nous les

avons analysés et classés selon le mécanisme de contention sur le canal utilisé par les nœuds capteurs dans le réseau en deux catégories suivantes: les premiers utilisent le mécanisme d'interrogation probabiliste tandis que les seconds utilisent le protocole *CSMA/CA* et par la suite une comparaison sous forme d'un tableau de ces protocoles est faite.

La seconde partie de la thèse fut consacrée à nos contributions dans le domaine de la gestion de l'énergie dans les *RCSF-RE*. Ces contributions se sont concrétisées, grâce à l'utilisation de la plate-forme de simulation *OMNeT++* version 4.6 avec le cadre de simulation *MiXiM* version 2.3. Le chapitre 4, présente le début de notre contribution dans lequel nous avons examiné deux types de protocoles *MAC* où les performances de conservation d'énergie sont évaluées. Le premier basé sur l'écoute à faible consommation d'énergie (*LPL*) tels que le *BMAC* et le *XMAC*, et le second sur le paradigme de balisage tel que *RIMAC*. Ces trois protocoles ont en commun l'introduire de la technique de duty-cycle afin de réduire les temps morts d'écoute, la principale cause de la consommation d'énergie. En général, les résultats de l'évaluation montrent que *RIMAC* conserve plus d'énergie que *XMAC* et *BMAC*, alors que le *XMAC* occupe la deuxième place devant *BMAC*.

En tant que contribution majeure à ce travail de thèse, nous avons conçu un nouveau schéma de gestion de l'énergie amélioré dans les *RCSF-RE* nommé *EEM-EHWSN* présenté dans le chapitre 5 où les nœuds de capteurs sont dotés d'un système de récupération d'énergie. *EEM-EHWSN* est un schéma *MAC* fonctionnant sur le paradigme de balisage par l'envoi d'une balise (*beacon*) dont la communication est initiée par le récepteur sans nécessiter de type de synchronisation entre les nœuds. Il permet aux nœuds de capteurs d'obtenir des gains importants en termes d'énergie et assure un équilibrage de charge dans le *RCSF*. *EEM-EHWSN* permet de réduire le duty-cycle par l'utilisation d'une politique de seuil d'énergie proposée et l'énergie résiduelle du nœud de capteur afin de réguler les deux périodes (réveil et veille). Pour ces dernières, premièrement : dans la période de réveil *EEM-EHWSN* utilise une technique de commutation où chaque nœud peut basculer entre deux modes (relais et de détection) pour réaliser l'équilibrage de charge, et deuxièmement : pour la période de veille, *EEM-EHWSN* calcule la durée de cette période en fonction de l'interpolation de Lagrange représentée dans l'équation 5.14 et de l'énergie restante en ajustant de manière dynamique leur temps de réveil suivant en fonction des conditions de récupération de l'énergie afin de réduire le temps de latence moyen et augmenter le débit et le taux de livraison de paquets dans le *RCSF*. De plus, *EEM-EHWSN* assure un équilibre entre la consommation d'énergie et la capacité de récupération d'énergie de chaque nœud de capteur dans le *RCSF*. Toutes ces fonctionnalités de *EEM-EHWSN* ont été décrites dans la figure 5.6. Dans toutes les expériences de simulation faite, notre schéma *EEM-EHWSN* surpasse de manière significative les autres schémas (les deux schémas *DSP* et *DSR* et le protocole *RIMAC*) en termes de taux de livraison des paquets, de latence moyenne de paquets et de débit. Dans le chapitre 6, nous avons proposé un autre nouveau protocole *MAC* basé sur un schéma d'échantillonnage de préambule pour les *RCSF-RE* appelé (*PS-*

EHWSN) afin de réaliser des communication dans les *RCSF-RE* dans le cas où il n'y a pas suffisamment de nœuds de transfert éveillés, soit parce que la densité des nœuds en service est trop basse ou encore que la période de récolte moyenne d'énergie est trop longue et qui représente un prélude à l'invention d'un autre protocole du type *DTN*. *PS-EHWSN* présente une nouvelle approche de communication initiée par l'émetteur qui utilise la technique d'écoute à faible consommation d'énergie (*LPL*) dans les *RCSF-RE* mobiles. *PS-EHWSN* utilise l'approche du préambule découpé en diffusant une série de paquets de préambule courts et fonctionne selon un schéma de planification dynamique du duty-cycle dont les durées des prochaines périodes de veille des nœuds sont calculées en utilisant leur niveau d'énergie résiduelle. Il contribue également à équilibrer la consommation d'énergie entre les nœuds de capteurs dans le *RCSF*. Le *PS-EHWSN* a été évalué avec de nombreux cas de tests de flux de trafic variable par comparaison avec le protocole *BMAC*. Dans toutes les expériences, *PS-EHWSN* surpasse le protocole *BMAC* en termes de taux de livraison, de latence moyenne et de débit des paquets.

2 Les perspectives

Pour surmonter l'énergie limitée des *RCSF* alimentés par batterie, l'énergie récupérée est considérée comme une solution potentielle pour la réalisation de systèmes autonomes. Un gestionnaire d'énergie (*GE*) est généralement intégré dans des nœuds sans fil pour adapter la charge de fonctionnement en modifiant leur intervalle de réveil et de veille en fonction de l'énergie récupérée. Dans cette thèse, afin de prolonger la durée de vie du réseau, le *GE* est assuré par l'utilisation d'une politique de seuil d'énergie dans le réseau. Les travaux effectués basés sur la récolte d'énergie dans cette thèse dégagent de nombreuses perspectives de recherche qui méritent d'être explorés que ce soit pour le système de gestion de l'énergie amélioré (*EEM-EHWSN*) ou pour le protocole *MAC* basé sur un schéma d'échantillonnage de préambule (*PS-EHWSN*) pour le *RCSF-RE*.

1. Pour gérer l'énergie dans le *RCSF-RE*, dans cette thèse nous avons utilisé une politique de seuil d'énergie pour le développement des deux schémas *EEM-EHWSN* et *PS-EHWSN*. Dans ce type de politique, les seuils d'énergie utilisés sont des constantes définis par l'administrateur et qui ne dépend pas de l'impact de plusieurs paramètres dans le *RCSF-RE* tels que : la quantité d'énergie récoltée, le type de source d'énergie utilisée, la mobilité des nœuds et les conditions du trafic dans le réseau. Ainsi, la définition de ces seuils d'énergie constantes ou variables en tenant compte de l'impact de ces paramètres cités précédemment qui peuvent être utilisés par tous les nœuds ou chaque nœud peut définir ses propres seuils d'énergie pourrait être envisagé afin de fournir une marge de possibilités; premièrement de récolter plus d'énergie et deuxièmement d'avoir suffisamment des nœuds éveillés pour assurer la communication dans le réseau. En outre, la définition de ces seuils peut être effectuée sur la base de l'utilisation des lois de probabilité telle que la loi de poisson.

2. Étant donné que la plupart des sources d'énergie ambiantes ont des caractéristiques diverses qui dépendent des différentes situations environnementales, ainsi une deuxième perspective est de proposer un algorithme du type *DTN (Delay Tolerant Network)*, dans le cas où il n'y a pas assez de nœuds de transport éveillés, soit parce que la masse des nœuds en service est trop basse ou encore que la période de récolte moyenne d'énergie est trop longue.
3. Une troisième perspective est d'implémenter nos deux schémas proposés (*EEM-EHWSN* et *PS-EHWSN*) à l'aide d'un autre simulateur tel que le *NS2* et les comparer avec d'autres travaux similaires sur la base des mêmes métriques de performances utilisées et avec d'autres métriques qui ne sont pas utilisées telles que : l'équité, la scalabilité, la qualité de service,...etc.
4. Enfin, une dernière idée concerne la connaissance de la vitesse de la récolte énergétique qui dépend de la source d'énergie utilisée après un apprentissage effectué pour cette dernière pourrait être une extension à nos travaux de recherche. Par exemple, la modélisation de cette vitesse afin d'améliorer la politique d'énergie proposée.

Aperçu sur la plate-forme de simulation OMNeT ++ et du modèle de simulation MiXiM

A.1 La Plate-forme de simulation OMNeT ++

OMNeT ++, est une bibliothèque et un cadre de simulation modulaire à base de composants C ++, principalement destinés à la construction de simulations de réseau. Cependant, il a été conçu pour être aussi général que possible. Il peut donc être utilisé avec succès dans d'autres domaines, tels que la simulation de systèmes informatiques complexes, les réseaux de files d'attente ou les architectures matérielles [Web 20]. Il fournit les outils et l'infrastructure nécessaires à la création de programmes de simulation, mais il ne s'agit pas d'un simulateur concret. Des fonctionnalités spécifiques à un domaine, telles que la prise en charge de RCSF ou de protocoles Internet, sont fournies par des cadres de modèles en tant que projets indépendants.

OMNeT ++ propose un environnement de développement intégré basé sur Eclipse et comprend à la fois une interface utilisateur graphique pour les exécutions interactives et animées et une interface de ligne de commande pour les exécutions par lots. Le cadre de simulation OMNeT ++ suit les principes de la simulation par événements discrets. OMNeT ++, fournit, entre autres, les fonctionnalités suivantes:

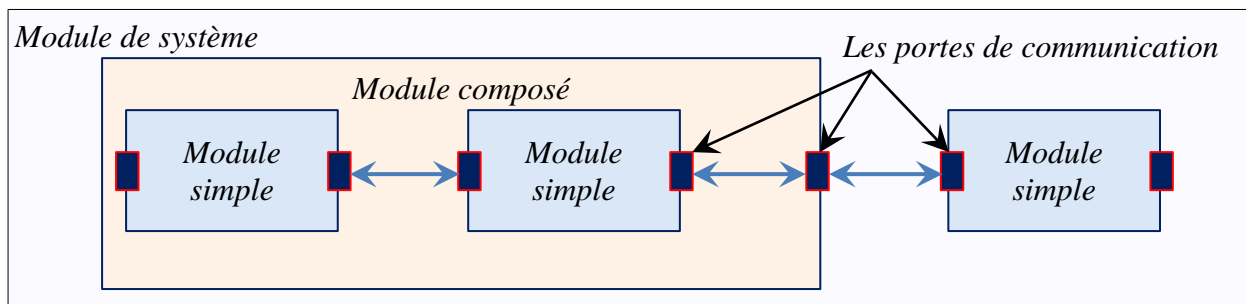


Figure A.1: Les différents modules d'un module système (simple et composé) [Web 20]

1. prise en charge des modules hiérarchiques imbriqués avec une profondeur illimitée :
 - 1- le module de niveau supérieur est appelé module système, qui contient un module simple de niveau inférieur ou un module composé contenant plusieurs modules simples, comme illustré à la figure A.1,
 - 2- les modules ont des portes comme interfaces d'entrée et de sortie,
 - 3- les modules communiquent en échangeant des messages;
2. langage de description de travail (*NED: Network Description*) pour la description de la topologie;

3. prise en charge de nombreuses distributions générant des nombres aléatoires;
4. prise en charge de la simulation parallèle;
5. collecte, analyse et visualisation de plusieurs types de résultats;
6. disponibilité de nombreux modèles de simulation tels que *INET Framework*, *INET-MANET*, *MiXiM*, *Castalia* et bien d'autres.
7. la séparation des expériences des modèles, où un modèle est défini par ses définitions *NED* avec le code *C++* correspondant. Les expériences sont définies dans les fichiers *INI* (*INItialization*). On suppose qu'un grand nombre d'expériences différentes doivent être menées sur un seul modèle avant qu'une conclusion puisse être tirée. Pour cette raison, bien que les paramètres puissent être codés dans les fichiers *NED*, il est recommandé de définir la plupart des paramètres à l'aide de fichiers *INI*.

A.2 Le modèle de simulation MiXiM

MiXiM est un modèle de simulation qui étend *OMNeT++* en prenant en charge la simulation de réseaux mobiles et sans fil. Il fournit des solutions en combinant les approches de plusieurs cadres de simulation existants en un seul: la prise en charge de la mobilité, la gestion des connexions et la structure générale sont tirées du cadre de mobilité (*MF: Mobility Framework*) [Web 21]; les modèles de propagation radio sont tirés du simulateur de chaîne (*ChSim*) [Web 22]; et la bibliothèque de protocoles provient du simulateur *MAC* [Web 23], du cadre positif [Web 24] et du cadre de mobilité. De plus, *MiXiM* introduit des extensions uniques telles que le support 3D intégral, des modèles d'obstacles et d'autres qui influencent la mobilité et l'atténuation des signaux radio, des fréquences et des supports de transmission différents (ondes radio, ultrasons); prise en charge multicanal dans l'espace et la fréquence, permettant des différentes simulations, en particulier la simulation des protocoles *MAC*, notamment *IEEE 802.15.4*. *MiXiM* est conçu pour prendre en charge la simulation de réseaux de plus de 1000 nœuds. Il consomme donc peu de mémoire et sa structure modulaire permet l'adaptation du niveau de détail et donc du temps d'exécution. Une interface de configuration graphique aide à choisir les bons modules, à les empiler en couches et à attribuer des valeurs à leurs paramètres. Afin de fournir les fonctionnalités indiquées, *MiXiM* contient de nombreux modules et interfaces *OMNeT++*. Nous en présentons ci-dessous quelques-unes qui sont pertinentes pour la mise en œuvre des cinq schémas *MAC* (*BMAC*, *XMAC*, *RIMAC*, *EEM-EHWSN* et *PS-EHWSN*).

Le module *ChannelControl* de *MiXiM*, en fonction de la distance et des caractéristiques physiques des nœuds, établit, maintient et détruit de manière dynamique toutes les connexions potentielles entre les nœuds. Ainsi, il permet à un nœud de recevoir chaque paquet de données que son émetteur peut détecter. La couche physique décide alors, en fonction de la force du signal reçu, s'il s'agit d'un paquet valide ou s'il doit être traité comme du bruit. Lors de l'initialisation, le module *ChannelControl* détermine la distance de brouillage maximale en fonction de paramètres tels que la fréquence porteuse du canal, la puissance de transmission des nœuds et d'autres paramètres spécifiques à la

propagation, puis calcule les connexions entre tous les nœuds et les met à jour à chaque fois que le changement se produit dans le réseau.

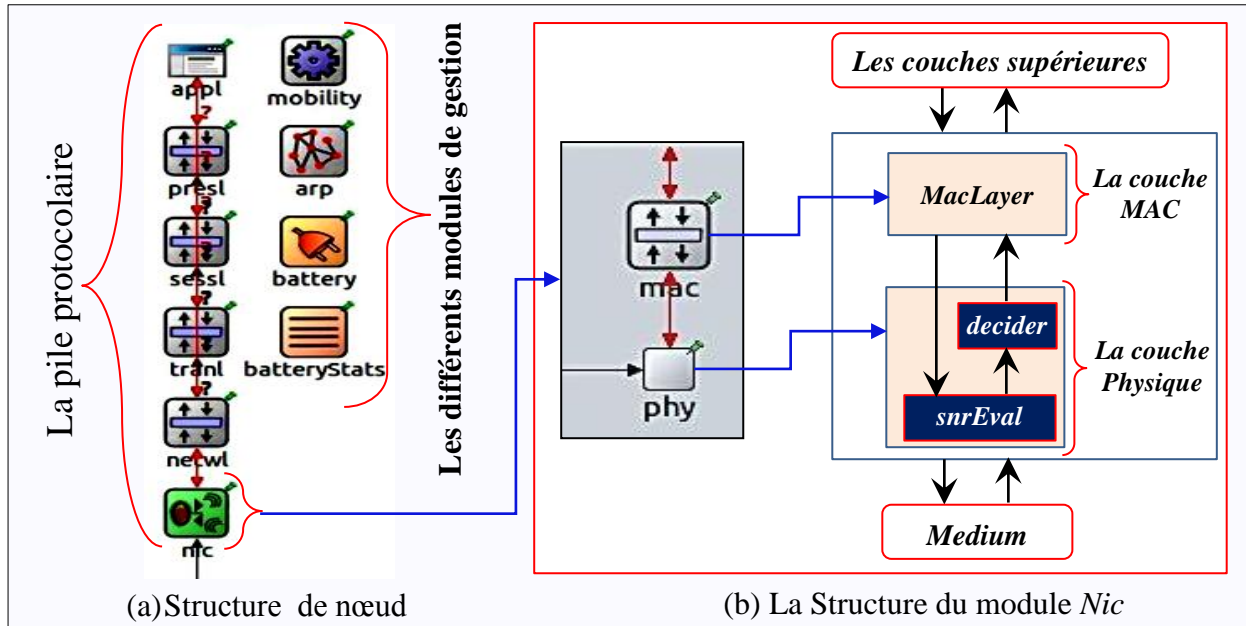


Figure A.2: La structure d'un nœud et la structure du module NIC [21]

Chaque couche de la pile de protocoles de communication, illustrée à la figure A.2(a), possède ses propres modules, qui assurent la fonctionnalité correspondante. Au niveau de la couche application, une application peut générer soit un, soit une rafale de messages de diffusion, qui sont ensuite transmis aux couches inférieures. Chaque couche ajoute les informations d'en-tête de la taille *headerLength* au paquet et les transmet à la couche suivante. La couche respective située à l'autre extrémité extrait les informations relatives à l'en-tête, puis supprime l'en-tête du paquet. Un module composé de carte d'interface réseau (NIC) décrit à la figure A.2(b) fournit les fonctionnalités des couches *PHY* (*physique*) et *MAC*. Au niveau de la couche *PHY*, il comporte deux sous-modules, à savoir un *snrEval* et un *décideur*, alors que le module de couche *MAC* est appelé *MACLayer*.

Le module *snrEval* simule le délai de transmission pour tous les messages reçus et calcule également les informations *SNIR*. Le module *décideur* peut utiliser ces informations pour décider si un message est perdu, présente des erreurs de bit ou est correctement reçu. Cependant, les messages reçus des couches supérieures sont directement ignorés dans *snrEval*. Les paramètres nécessaires au calcul du *SNIR* et de l'atténuation d'un signal sont *transmitterPower*, *carrierFrequency* et *pathLossAlpha*. Notez que le module *ChannelControl* possède également des versions de ces paramètres (*pMax*, *carrierFrequency*, *alpha*), mais elles ne sont utilisées que pour calculer la distance d'interférence des nœuds. Un utilisateur peut définir la valeur du seuil d'atténuation du signal (*SAT*) pour les niveaux de puissance reçus du signal requis. Un signal plus faible que *SAT* est simplement négligé. Le module *MACLayer* fonctionne en étroite coordination avec *snrEval* et *décideur* et permet à un nœud de détecter la porteuse et de communiquer de manière fiable avec les autres nœuds.

En outre, *MiXiM* possède également un module *Blackboard* qui fournit une fonctionnalité de couches croisées en offrant un échange d'informations entre différentes couches. Ces informations peuvent inclure l'état de la radio, le niveau d'énergie actuel ou l'apparence d'affichage d'un nœud. Par exemple, un protocole *MAC* lors de la détection de la porteuse interroge le tableau sur l'état de la porteuse auprès du *PHY* ou lui permet de distinguer, à l'aide du module *Blackboard*, si la radio est en mode veille, réception ou transmission.

1- Références bibliographiques

1. Chiang, S. Y., Kan, Y. C., Chen, Y. S., Tu, Y. C., & Lin, H. C. « Fuzzy Computing Model of Activity Recognition on WSN Movement Data for Ubiquitous Healthcare Measurement ». *Sensors*, 16(12), 2053. (2016)
2. Akhtar, R., Leng, S., & Memon, I. « Architecture for efficient content distribution in hybrid mobile social networks ». *Control Engineering and Electronics Engineering*, 95, 399-409. (2014)
3. Akhtar, R., Leng, S., Memon, I., Ali, M., & Zhang, L. « Architecture of hybrid mobile social networks for efficient content delivery ». *Wireless Personal Communications*, 80(1), 85-96. (2015)
4. Chen, M., Gonzalez, S., Vasilakos, A., Cao, H. & C. M. Leung, V. « Body Area Networks: A Survey. *Mobile Networks and Application* ». *J*, Aug, 16(2):171-193. (2010)
5. Ahmed, H. I., Wei, P., Memon, I., Du, Y., & Xie, W. « Estimation of time difference of arrival (TDoA) for the source radiates BPSK signal ». *IJCSI International Journal of Computer Science Issues*, 10(3), 164-171. (2013)
6. Memon, I., & Arain, Q. A. « Dynamic path privacy protection framework for continuous query service over road networks ». *World Wide Web*, 20(4), 639-672. (2017)
7. Memon, I., Chen, L., Arain, Q. A., Memon, H., & Chen, G. « Pseudonym changing strategy with multiple mix zones for trajectory privacy protection in road networks ». *International Journal of Communication Systems*, 31(1), e3437. (2017)
8. Ali, N. A., ElSayed, H. M., El-Soudani, M., Amer, H. H., & Daoud, R. M. « Elongation of WSN lifetime using a centralized clustering technique ». *International Journal of Systems, Control and Communications*, 4(4), 250–261. (2012)
9. Batra, P. K., & Kant, K. « A clustering algorithm with reduced cluster head variations in LEACH protocol ». *International Journal of Systems, Control and Communications*, 7(4), 321–336. (2016)
10. Fan, C. S. « An energy-efficient two phases cluster head selection in corona-based wireless sensor networks ». *Int. J of Ad Hoc and Ubiquitous. Computing*, 20(1), 17–25. (2015)
11. Chuang, P. J., Yang, S. H., & Lin, C. S. « Energy-efficient clustering in wireless sensor networks ». In A. Hua, S. L. Chang (Eds.), *Algorithms and architectures for parallel processing. ICA3PP 2009. Lecture Notes in computer science*, (Vol. 5574, pp. 112–120), Berlin: Springer. (2009)

12. Kim, H. Y. « An energy-efficient load balancing scheme to extend lifetime in wireless sensor networks ». *International Journal of Cluster Computing*, 19(1), 279–283. (2016)
13. Arain, Q. A., Uqaili, M. A., Deng, Z., Memon, I., Jiao, J., Shaikh, M. A., & al. « Clustering based energy efficient and communication protocol for multiple mix-zones over road networks ». *Wireless Personal Communications*, 95(2), 411–428. (2017)
14. Maheswar, R., Jayarajan, P., & Sheriff, F. N. « A survey on duty cycling schemes for wireless sensor networks ». *International Journal of Computer Networks and Wireless Communications*, 3(1), 37–40. (2013)
15. Sudevalayam, S., & Kulkarni, P. « Energy harvesting sensor nodes: Survey and implications ». *IEEE Communications Surveys & Tutorials*, 13(3), 443–461. (2011)
16. Ma, S., Yang, Y., Qian, Y., Sharif, H., & Alahmad, M. « Energy harvesting for wireless sensor networks: Applications and challenges in smart grid ». *International Journal of Sensor Networks*, 21(4), 226–241. (2016)
17. Jelic'ic'. V. « Power management in wireless sensor networks with high-consuming sensors ». Qualifying Doctoral Examination. (2011)
18. Polastre, J., Hill, J., & Culler, D. « Versatile Low Power Media Access for Wireless Sensor Networks ». In *SenSys'04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, (New York, NY, USA), pp. 95–107, ACM. (2004)
19. Buettner, M., Yee, G. V., Anderson, E., & Han, R. « XMAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks ». In *SenSys'06: Proceedings of the 4th international conference on Embedded networked sensor systems*, (New York, NY, USA), pp. 307–320, ACM. (2006).
20. Sun, Y., Gurewitz, O., & Johnson, D. B. « RIMAC: A Receiver Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks ». In *SenSys '08: Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems*. (2008)
21. Köpke, A., Swigulski, M., Wessel, K., Willkomm, D., Haneveld, P. T. K., Parker, T. E. V., & al. « Simulating wireless and mobile networks in omnet++ the mixim vision ». In *Proceedings of the 1st international conference on simulation tools and techniques for communications, networks and systems (SIMUTools)*. Marseille, France: ICST (pp. 71:1–71:8). (2008)
22. **Bengheni, A.**, Didi, F., & Bambrik, I. « Energy-harvested management mechanism for Wireless Sensor Networks ». *The 5th International Conference on Electrical Engineering – Boumerdes (ICEE-B)*, Algeria, October 29th-31th. (2017)
23. Dhoutaut, D. « Etude du standard *IEEE* 802.11 dans le cadre des réseaux *Ad Hoc* : de la simulation à l'expérimentation ». Thèse de doctorat, L'Institut National des Sciences Appliquées de Lyon, Décembre. (2003)
24. Fette, B., Aiello, R., Chandra, P., Dobkin, D. M., Bensky, A., Miron, D., Lide, D. A., Dowla, F., & Olexa, R., « *RF and Wireless Technologies* ». Elsevier-Newness. (2008)

25. Ilyas, M. « The Handbook of Ad Hoc Wireless Networks ». USA, CRC Press LLC. (2003)
26. Christian, B. « On the Connectivity of Ad Hoc Networks ». In the Computer Journal, 47(4), 432-447. (2004)
27. Winston, J. J., & Paramasivan, B. « A Survey on Connectivity Maintenance and Preserving Coverage for Wireless Sensor Networks ». International Journal of Research and Reviews in WSN (IJRRWSN), 1(2), 11-18, June. (2011)
28. Lynch, J. P., Sundararajan, A., Law, K. H., Kiremidjian, A. & Carryer, S., E. « Embedding damage detection algorithms in a wireless sensing unit for operational power efficiency ». Smart Materials and Structures, 13(4), 800-810, April. (2004)
29. Intanagonwiwat, C., Govindan, R., & Estrin, D. « Directed Diffusion: a Scalable and Robust Communication Paradigm for Sensor Networks ». ACM MobiCom, Boston, MA. (2000)
30. Karl, H., & Willig, A. « Protocols and Architectures for Wireless Sensor Networks ». John Wiley and Sons Ltd. (2006)
31. Karl, H., & Willig, A. « Protocols and Architectures for Wireless Sensor Networks ». Wiley, 5, 6, 7, 14, 20, 21. (2005)
32. Matin, M.A., & Islam, M.M. « Overview of Wireless Sensor Networks. Creative Commons Attribution License ». (2012)
33. Akyildiz, I. F., Weilian, S., Sankarasubramaniam, Y., & Cayirci, E. « A survey on sensor networks ». Communications Magazine, IEEE, 40(8), 102-114, August. (2002)
34. Davis, A., & Chang, H. « A survey of Wireless Sensor Networks Architectures ». International Journal of Computer Science & Engineering Survey (IJCSSES) 3(6), December. (2012)
35. Watteyne, T. « Proposition et validation formelle d'un protocole MAC temps réel pour réseaux de capteurs linéaires sans fils ». Mémoire de Master Recherche Informatique, Laboratoire CITI, Lyon, France. (2005)
36. Chong, C.Y., & Kumar, S.P. « Sensor Networks: Evolution, Opportunities, and Challenges ». In Proceedings of the IEEE, 91(8), 1247- 1256, August. (2003)
37. Kumar, S., & Shepherd, D. « SensIT: Sensor information technology for the warfighter ». In Proceedings of the. 4th International Conference on Information Fusion, pages 1-6. (2001)
38. Johnson, P., & Andrews, D.C. « Remote continuous monitoring in the home ». Journal of Telemedicine and Telecare, 2(2), 107-113, June. (1996)
39. Schott, W., Gluhak, A., Presser, M., Hunkeler, U., & Tafazolli, R. « e-SENSE Protocol Stack Architecture for Wireless Sensor Networks ». In 16th IST on Mobile and Wireless Communications Summit, pp. 1-5, July. (2007)
40. Abu-Ghazaleh, N., Kang, K. D., & Liu, K. « Towards resilient geographic routing in WSNs ». In Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks, pp. 71-78, October. (2005)

41. Raghunathan, V., Schurgers, C., Park, S., & Srivastava, M.B. « Energy-Aware Wireless Microsensor Networks ». *IEEE Signal Processing Magazine*, 19(2), 40-50, August. (2002)
42. Shih, E., & al. « Physical Layer driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks ». *Proc. ACM MobiCom '01*, pp. 272–287, Rome, Italy, July. (2001)
43. Heinzelman, W., Chandrakasan, A., & Balakrishnan, H. « Energy-efficient communication protocol for wireless sensor networks ». *Proceedings of the IEEE Hawaii International Conference on System Sciences*, pp. 3005-3014, January. (2000)
44. Pottie, G. J. & Kaiser, W. J. « Wireless Integrated Network Sensors ». *Communication of the ACM*, 43(5), 51-58, May. (2000)
45. Akyildiz, I. F., & Vuran, M. C. « Wireless Sensor Networks ». John Wiley and Sons Ltd. (2010)
46. Bachir, A., Dohler, M., Watteyne, T., & Leung, K.K. « MAC Essentials for Wireless Sensor Networks ». *IEEE Communications Surveys and Tutorials*, 12(2), 222-248, April. (2010)
47. Chen, Y., & Zhao, Q. « On the Lifetime of Wireless Sensor Networks ». *IEEE Communication Letters*, 9(11), 976-978, November. (2005)
48. Dietrich, I., & Dressler, F. « On the Lifetime of Wireless Sensor Networks ». *ACM Transactions on Sensor Networks (TOSN)*, 5(1), February. (2009)
49. Mak, N. H., & Seah, W. K. G. « How Long is the Lifetime of a Wireless Sensor Network? ». In *International Conference on Advanced Information Networking and Applications (AINA'09)*, May. (2009)
50. Zhang, H.H., & Hou, J.C. « Maintaining Sensing Coverage and Connectivity in Large Sensor Networks ». *Ad Hoc & Sensor Wireless Networks*, 1(1), 89-124, March. (2005)
51. ISO/IEC. « Reference Model of Open Systems Interconnection; Part 1, Basic Reference Model (incorporating connectionless-mode transmission), ». tech. rep., British Standards Institution, 2 Park Street, London W1A 2BS, UK. BS 6568: Part 1: 1988 (ISO 7498–1984 including Amendment 1). (1988)
52. Khan, J. A., Qureshi, H. K., & Iqbal, A. « Energy management in Wireless Sensor Networks: A survey ». *Computers and Electrical Engineering*, Elsevier, 41, January, 159-176. (2015)
53. Tudose, D., & Tapus, N. « Energy Harvesting and Power Management in Wireless Sensor Networks ». *18th International Conference of Control Systems and Computer Science CSCS18.*, Bucharest, Romania, May, 1, 174-180. (2011)
54. **Bengheni, A.**, Didi, F., & Bambrik, I. « Energy-saving comparaison of asynchronous MAC protocols for WSNs ». *The International Conference on Mathematics and Information Technology (ICMIT)*, Adrar, Algeria, December 04th-05th. (2017)

55. Anastasi, G., Francesco, M.D., Conti, M., & Passarella, A. « How to Prolong the Lifetime of Wireless Sensor Networks ». *Mobile Ad Hoc and Pervasive Communications*, pp. 1–26. (2006)
56. Schurgers, C., Tsiatsis, V., & Srivastava, M.B. « Stem: Topology management for energy efficient sensor networks ». In *Proceedings of the IEEE Aerospace Conference '02*, Big Sky, MT, March 10-15. (2002)
57. Zheng, R., Hou, J.C., & Sha, L. « Asynchronous wakeup for ad hoc networks ». In *Proceedings of the 4th ACM international symposium on Mobile ad Hoc networking & computing (MobiHoc'03)*, pages 35–45, New York, NY, USA. ACM. (2003)
58. Klues, K., Hackmann, G., Chipara, O., & Lu, C. « A Component Based Architecture for Power-Efficient Media Access Control in Wireless Sensor Networks ». In *The 5th International Conference on Embedded Networked Sensor Systems (SenSys'07)*, pages 59-72, November (2007)
59. Dargie, W., & Poellabauer, C. « *Fundamentals of Wireless Sensor Networks: Theory and Practice* ». John Wiley and Sons Ltd. (2010)
60. Zheng, J., & Jamalipour, A. « *Wireless Sensor Networks, A Networking Perspective* ». John Wiley and Sons, Inc., Hoboken, New Jersey. (2009)
61. I. 802.11. « Carrier Sense Multiple Access with Collision Detection CSMA/CD access method and physical layer specifications ». ANSI/IEEE, Standard 802.3. (1998)
62. Mühlethaler, P. « 802.11 et les réseaux sans fil. Eyrolles ». (2002)
63. Ye, W., Heidemann, J., & Estrin, D. « Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks ». *IEEE/ACM Trans. Net.*, 12(3): 493-506, June. (2004)
64. Dam, V. T., & Langendoen, K. « An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks ». In *1st ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 171-180, November. (2003)
65. Lu, G., Krishnamachari, B., & Raghavendra, C. « An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks ». In *Ad Hoc and Sensor Networks (WMAN)*, April. (2004)
66. Cheng, P., Zhang, F., Chen, J., Sun, Y., & Shen, X. « A Distributed TDMA Scheduling Algorithm for Target Tracking in Ultrasonic Sensor Networks ». *IEEE Transactions on Industrial Electronics*, 60(9), 3836–3845, Sep. (2013)
67. Rajendran, V., Obraczka, K., & Garcia-Luna-Aceves, J. J. « Energy-efficient Collision-free Medium Access Control for Wireless Sensor Networks ». In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, pp. 181–192. (2003)
68. Rajendran, V., Garcia-Luna-Aceves, J. J., & Obraczka, K. « Energy-efficient, application-aware medium access for sensor networks ». In *IEEE International Conference on Mobile Ad hoc and Sensor Systems Conference*, November. (2005)

69. Zhang, F., Chen, J., Li, H., Sun, Y., & (Sherman) Shen, X. « Distributed Active Sensor Scheduling for Target Tracking in Ultrasonic Sensor Networks ». *Mobile Netw Appl*, 17(5), 582–593, Oct. (2012)
70. Bachir, A., Barthel, D., Heusse, M., & Duda, A. « Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks ». In *IEEE International Conference on Communications, ICC'06*, 7, 3365–3370. (2006)
71. El-Hoiydi, A. & Decotignie, J.D. « WiseMAC : An Ultra-Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks ». In *Proceedings of the 9th International Symposium on Computers and Communications, ISCC 2004*, Vol 1. (2004)
72. Rhee, I., Warrier, A., Aia, M., Min, J., & Sichitiu, M. L. « ZMAC: A Hybrid MAC for Wireless Sensor Networks ». *IEEE/ACM Transactions on Networking*, 16(3), 511–524, Jun. (2008)
73. Ye, W., Silva, F., & Heidemann, J. « Ultra-low Duty Cycle MAC with Scheduled Channel Polling ». In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, pp. 321-334, October. (2006)
74. Ahn, G.-S., Hong, S. G., Miluzzo, E., Campbell, A. T., & Cuomo, F. « Funneling-MAC: A Localized, Sink-oriented MAC for Boosting Fidelity in Sensor Networks ». In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, pp. 293–306. (2006)
75. Oudani, H., Krit, S. D., EL Maimouni, L., & Laassiri, J. « Energy consumption in wireless sensor network: simulation and comparative study of flat and hierarchical routing protocols ». *IADIS International Journal on Computer and Information Systems*, 12(1), 109-125, March. (2017)
76. **Bengheni, A.**, & Didi, F. « Improvement of the Lifetime of Wireless Sensor Network: state of the Art ». *The second International Workshop on Mathematics and Computer Science IWMCS' 2014*, Tiaret-Algeria, December 1th- 2th. (2014)
77. Goel, S., & Imielinski, T. « Prediction-based monitoring in sensor networks: taking lessons from mpeg ». *ACM SIGCOMM Computer Communication Review*, 31(5), 82–98. (2001)
78. Marcelloni, F., & Vecchio, M. « A simple algorithm for data compression in wireless sensor networks ». *Communications Letters, IEEE*, 12(6), 411-413. (2008)
79. Petrovic, D., Shah, R.C., Ramchandran, K., & Rabaey, J. « Data funneling: Routing with aggregation and compression for wireless sensor networks ». In: *Sensor Network Protocols and Applications*, 2003. *Proceedings of the First IEEE. 2003 IEEE Inter-national Workshop on. IEEE.* pp. 156–162. (2003)
80. Pradhan, S.S., Kusuma, J., & Ramchandran, K. « Distributed compression in a dense microsensor network ». *Signal Processing Magazine, IEEE*, 19(2), 51–60. (2002)
81. Yao, Y., & Gehrke, J. « The cougar approach to in-network query processing in sensor networks ». *ACM Sigmod Record*, 31(3), 9–18. (2002)

82. Anastasi, G., Conti, M., Passarella, A., & Pelusi, L. « Mobile-relay forwarding in opportunistic networks ». In M. Ibnkahla, editor, *Adaptation and Cross Layer Design in Wireless Networks*. CRC Press, New York, USA, 31. (2008)
83. **Bengheni, A.**, Didi, F., & Bambrik, I. « *EEM-EHWSN*: Enhanced Energy Management Scheme in Energy Harvesting WSNs ». *Wireless Networks (The Journal of Mobile Communication, Computation and Information Springer, ISSN 1022-0038)*, March, (2018)
84. Kansal, A., Hsu, J., Zahedi, S., & Srivastava, M. B. « Power management in energy harvesting sensor networks ». In *ACM Trans. on Embedded Computing Sys.*, 6(4), pp. 32, ACM, New York, NY, USA, December (2007)
85. Moser, C. « Power Management in Energy Harvesting Embedded Systems ». Phd Dissertation, SWISS Federal Institute Of Technology. (2009)
86. Basagni, S., Nderi, Y., Petrioli, C., & Spenza, D. « wireless sensor networks with energy harvesting ». John Wiley and Sons Inc, Hoboken. (2012)
87. Chalasani, S., & Conrad, J. M. « A survey of energy harvesting sources for embedded systems ». In *Proc. IEEE SoutheastCon*, Huntsville, April, pp. 442–447. (2008)
88. Roundy, S., Steingart, D., Wright, L., & Rabaey, J. « Power sources for wireless sensor networks ». *Proc. IEEE EWSN*. (2004)
89. Zhang, J., Luo, X., Chen, C., Liu, Z., & Cao, S. « A wildlife monitoring system based on wireless image sensor networks ». *Sensors Transducers* 180, 10 (2014), 104–109. (2014)
90. Zhang, P., Sadler C. M., Lyon, S. A., & Martonosi, M. « Hardware Design Experiences in ZebraNet ». In *Proceedings of the Second International Conference on Embedded Networked Sensor Systems*, pages 227–238. ACM. (2004)
91. Morelli, X., Rieux, C., Cyrus, J., Forsberg, B., & Slama, R. « Air pollution, health and social deprivation: A fine-scale risk assessment ». *Environ. Res.* 147, 59–70. (2016)
92. Touati, F., Galli, A., Crescini, D., Crescini, P., & Mnaouer, A. B. « Feasibility of air quality monitoring systems based on environmental energy harvesting ». In *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference (I2MTC'15)*. IEEE, Pisa, Italy, 266–271. (2015)
93. Wang, C., Huang, Y., Zheng, X., Lin, T., Chuang, C., & Jiang, J. « A self-sustainable air quality monitoring system using WSN ». In *Proceedings of the 5th IEEE International Conference on Service-Oriented Computing and Applications (SOCA'12)*. IEEE, 1–6. (2012)
94. Chottirapong, K., Manatrinon, S., Dangsakul, P., & Kwankeow, N. « Design of energy-harvesting thermoelectric generator with wireless sensors in organic fertilizer plant ». In *Proceedings of the 6th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES'15)*. IEEE, Hua-Hin, Thailand, 1–6. (2015)
95. Wang, T. « Water Quality Monitoring System based on WSN ». Master's thesis. KTH Royal Institute of Technology, Sweden. (2012)

96. Cavallari, R., Martelli, F., Rosini, R., Buratti, C., & Verdone, R. « A survey on wireless body area networks: Technologies and design challenges ». *IEEE Commun. Surveys Tutor.* 16(3), 1635–1657. (2014)
97. Noel, A., Abdaoui, A., Badawy, A., Elfouly, T., Ahmed, M., & Shehata, M. « Structural health monitoring using wireless sensor networks: A comprehensive survey ». *IEEE Commun. Surveys Tutor.* (2017)
98. Akhtar, F. & Rehmani, M. H. « Energy harvesting for self-sustainable wireless body area networks ». *IT Profess.* 19(2), 32–40. (2017)
99. Saida, M., Zaibi, G., Samet, M., & Kachouri, A. « Improvement of energy harvested from the heat of the human body ». In *Proceedings of the 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA'16)*. IEEE, 132–137. (2016)
100. Leonov, V. « Thermoelectric energy harvesting of human body heat for wearable sensors ». *IEEE Sensors J.* 13(6), 2284–2291. (2013)
101. Borges, L. M., Chávez-Santiago, R., Barroca, N., Velez, F. J., & Balasingham, I. « Radio-frequency energy harvesting for wearable sensors ». *Healthcare Technol. Lett.* 2(1), 22–27. (2015)
102. Xia, L., Cheng, J., Glover, E. N., & Chiang, P. « 0.56V, –20 dBm RF-powered, multi-node wireless body area network system-on-a-chip with harvesting-efficiency tracking loop ». *IEEE J. Solid-State Circuits* 49(6), 1345–1355. (2014)
103. Rasaneh, S. & Banirostan, T. « A new structure and routing algorithm for optimizing energy consumption in wireless sensor network for disaster management ». In *Proceedings of the 4th International Conference on Intelligent Systems, Modelling and Simulation*. IEEE, 481–485. (2013)
104. Hande, A., Bridgelall, R., & Zoghi, B. « Vibration energy harvesting for disaster asset monitoring using active RFID tags ». *Proc. IEEE* (Sept), 98(9), 1620–1628. (2010)
105. Mihajlovic, Z., Joza, A., Milosavljevic, V., Rajs, V., & Zivanov, M. « Energy-harvesting wireless sensor node for monitoring of surface water ». In *Proceedings of the 21st International Conference on Automation and Computing (ICAC'15)*. IEEE, 1–6. (2015)
106. Lung, C., Sabou, S., & Buchman, A. « Wireless sensor networks as part of emergency situations management system ». *IEEE 22nd International Symposium for Design and Technology in Electronic Packaging (SIITME)*. (2016)
107. Antolín, D., Medrano, N., Calvo, B., & Pérez, F. « A Wearable Wireless Sensor Network for Indoor Smart Environment Monitoring in Safety Applications ». *Sensors*, 17(2), pii: E365. (2017)
108. Eu, Z. A., Tan, H. P., & Seah, W. K. G. « Design and performance analysis of MAC schemes for wireless sensor networks powered by ambient energy harvesting ». *Ad-Hoc Network*, 9(3), 300–323. (2011)
109. Eu, Z. A., & Tan, H. P. « Probabilistic polling for multihop energy harvesting wireless sensor networks ». In *IEEE Interenational Symposium on Adhoc Sensor Network*, Ottawa, Canada, June 10th–15th (pp. 271–275). (2012)

110. Fujii, C., & Seah, W. K. G. « Multi-tier probabilistic polling in wireless sensor networks powered by energy harvesting ». IEEE international conference on intelligent sensors, sensor network. Information process, Adelaide, Australia, Dec 6th–9th (pp. 383–388). (2011)
111. Fafoutis, X., & Dragoni, N. « ODMAC: An on-demand MAC protocol for energy harvesting wireless sensor networks ». In Proceedings of 8th ACM symposium on performance evaluation of wireless ad-hoc, sensor, and ubiquitous network, Miami, FL, USA (pp. 49–56). (2011)
112. Chuang, P. J., Yang, S. H., & Lin, C. S. « Energy-efficient clustering in wireless sensor networks ». In A. Hua, S. L. Chang (Eds.), Algorithms and architectures for parallel processing. ICA3PP 2009. Lecture Notes in computer science (Vol. 5574, pp. 112–120). Berlin: Springer. (2009)
113. Nguyen, K., Nguyen, V. H., Le, D. D., Ji, Y., Duong, D. A., & Yamada, S. « ERI-MAC: An energy harvested receiver initiated MAC protocol for wireless sensor networks ». International Journal of Distributed Sensor Networks, 1th–8th. (2014)
114. Kim, S. C., Jeon, J. H., & Park, H. J. « QoS aware energyefficient (QAEE) MAC protocol for energy harvesting wireless sensor networks ». In Convergence hybrid information, technology, Daejeon, Republic of Korea (pp. 41–48). (2012)
115. Ramezani, P., & Pakravan, R. M. « Overview of MAC protocols for energy harvesting wireless sensor networks ». In IEEE 26th international symposium on personal, indoor and mobile radio communications-(PIMRC): Mobile and wireless networks (pp. 2032–2037). (2015)
116. Layerle, D., & Kwasinski, A. « A power efficient pulsed mac protocol for body area networks ». In IEEE 22nd international symposium on personal indoor and mobile radio communications (PIMRC), Tronto, ON, Canada, Sept 11th–14th (pp. 2244–2248). (2011)
117. Ali, N. A., ElSayed, H. M., El-Soudani, M., Amer, H. H., & Daoud, R. M. « Elongation of WSN lifetime using a centralised clustering technique ». International Journal of Systems, Control and Communications, 4(4), 250–261. (2012)
118. Kim, Y., Park, C. W., & Lee, T. J. « MAC protocol for energy harvesting users in cognitive radio networks ». In: Proceedings of 8th international conference on ubiquitous information management and communication. (2014)
119. Liu, H. I., He, W. J., & Seah, W. K. G. « LEBMAC: Load and energy balancing MAC protocol for energy harvesting powered wireless sensor networks ». In 20th IEEE international conference on parallel and distributed systems (ICPADS), Hsinchu, Taiwan. (2014)
120. Lin, H. H., Shih, M. J., Wei, H. Y., & Vannithamby, R. « DeepSleep: IEEE 802.11 enhancement for energy-harvesting machine-to-machine communications ». Wireless Networks, 21(2), 357–370. (2014)
121. Yoo, H., Shim, M., & Kim, D. « Dynamic duty-cycle scheduling schemes for energy-harvesting wireless sensor networks ». IEEE Communications Letters, 16(2), 202–204. (2012)

122. Jain, R. « The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling ». John Wiley and Sons Inc., New York, US, April (1991)
123. Wehrle, K., Günes, M., & Gross, J. « Modeling and Tools for Network Simulation ». Springer Verlag, (2010)
124. **Bengheni, A.**, Didi, F., & Bamrik, I. « A Preamble Sampling Scheme based MAC Protocol for Energy Harvesting Wireless Sensor Networks ». The 3rd Conference on Computing Systems and Applications (CSA), Ecole Militaire Polytechnique, Algiers, Algeria, April 24th-25th, (2018)
125. Halkes, G.P., Van Dam, T. & Langendoen, K.G. « Comparing Energy-saving MAC Protocols for Wireless Sensor Networks ». Mobile Networks and Applications, vol. 10, pp. 783-791, (2005)
126. Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, R., Scott, K. & Weiss, H. « Delay-tolerant networking: an approach to interplanetary Internet». IEEE communication Magazine, 41(06), 128-136. (2003)
127. Baronti, P., Pillai, P., Chook, V., Chessa, S., Gotta, A. & Hu Fun, Y. «Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards ». Computer communication (Elsevier), 30(7), 1655-1695, May. (2007)
128. Bickerstaffe, J., « Energy Harvesting, SAGENTIA, Steve Beeby », in Energy Harvesting Systems, published by Springer, © Sagentia. (2011)

2- Webographie

- Web 01. « MiXiM Documentation », accessible via le lien: <http://mixim.sourceforge.net/>
Consulté le 07/05/2015
- Web 02. « Chipcon Radios », accessible via le lien: <http://focus.ti.com/analog/docs/enggres/detail.tsp?familyId=367&genContentId=3573>
Consulté le 07/10/2017
- Web 03. « MPR2400 Radio », accessible via le lien: http://www.willow.co.uk/html/mpr2400_micz_zigbee.html
Consulté le 10/10/2017
- Web 04. « RFM TR Radio », accessible via le lien: <http://www.rfm.com/products/data/tr3000.pdf>
Consulté le 10/10/2017
- Web 05. « Wireless Sensors Platform c, Intel Platform (2010) », accessible via le lien: <http://www.xbow.com/Products/productdetails.aspx?sid=280>
Consulté le 10/10/2017
- Web 06. « Wireless Sensors Platform, WSN tmoteSky Platform specifications », accessible via le lien: <http://www.sentilla.com/moteiv-transition.html>
Consulté le 10/10/2017
- Web 07. « TinyNode User's Manual », accessible via le lien: http://www.tinynode.com/uploads/media/TinyNode_Users_Manual_rev11.pdf
Consulté le 12/11/2017
- Web 08. « BTnodes - A Distributed Environment for Prototyping Ad Hoc Networks », accessible via le lien: <http://www.btnode.ethz.ch/>
Consulté le 12/11/2017
- Web 09. « An EYES sensor node », accessible via le lien: <http://www.eyes.eu.org/>
Consulté le 12/11/2017
- Web 10. « ZigbeX mote (Corée du sud) », accessible via le lien: <http://www.hanback.com/english/sub1.htm>
Consulté le 12/11/2017
- Web 11. « Scatterweb (Allemagne) », accessible via le lien: <http://www.scatterweb.com>
Consulté le 12/11/2017
- Web 12. « SensorScope », accessible via le lien: <http://lcav.epfl.ch/sensorscope-en>,
Consulté le 05/01/2018
- Web 13. « XBOW, produit et technologie des RCSF », accessible via le lien: <http://www.xbow.com/>
Consulté le 05/01/2018
- Web 14. « T. Armstrong, Wakeup Based Power Management in Multi-hop Wireless Networks », accessible via le lien: <http://www.eecg.toronto.edu/trevor/Wakeup/index.html>
Consulté le 18/01/2018
- Web 15. M. Raju and M. Grazier, « Ultra Low Power Meets Energy Harvesting, 2010 », accessible via le lien: www.ti.com/lit/wp/slyy018a/slyy018a.pdf
Consulté le 18/01/2018
- Web 16. NICS labs, « Critical infrastructures protection, 2014 », accessible via le lien: <https://www.nics.uma.es/research/critical-infrastructures-protection>
Consulté le 18/01/2018
- Web 17. A. F. Pele, « energy harvesting solutions, 2012 », accessible via le lien: <http://www.embedded.com/print/4403423>
Consulté le 20/03/2018
- Web 18. « TurtleNet », accessible via le lien: <http://prisms.cs.umass.edu/dome/turtlenet>
Consulté le 17/04/2018
- Web 19. « Texas Instruments. 2013b », accessible via le lien: <http://www.ti.com/lit/ds/symlink/msp430g2553.pdf>
Consulté le 30/04/2018
- Web 20. « La Plate-forme de simulation OMNeT ++ », accessible via le lien: <http://www.omnetpp.org/>
Consulté le 30/04/2018

- Web 21. « Mobility framework (MF) for simulating wireless and mobile networks using OMNeT++ », accessible via le lien: <http://mobility-fw.sourceforge.net/>
Consulté le 13/05/2018
- Web 22. S. Valentin. ChSim –« a wireless channel simulator for OMNeT++ », TKN Simulation Workshop 2006, Technical University of Berlin, Germany, Sept. 2006, accessible via le lien: <http://wwwcs.upb.de/cs/chsim>.
Consulté le 17/05/2018
- Web 23. « MAC simulator », accessible via le lien: <http://www.consensus.tudelft.nl/software.html>
Consulté le 25/05/2018
- Web 24. « Positif localization simulation framework », accessible via le lien: [http://www.consensus.tudelft.nl /software.html](http://www.consensus.tudelft.nl/software.html)
Consulté le 30/05/2018

إن حصاد الطاقة هي عملية يتم فيها جمع الطاقة من مصادر خارجية (مثل: ضوء الشمس، الضوء الإصطناعي، الحركة الاهتزازية والحرارة ، وغيرها...) وتحول هذه الطاقة إلى طاقة كهربائية يمكن إستخدامها مباشرة أو تخزينها قصد صرفها في تشغيل أجهزة الصغيرة. كما يمكن إستخدام هذا الحصاد من الطاقة المحيطة لتشغيل أجهزة مستشعرات لاسلكية صغيرة مستقلة ذات بطاريات صغيرة أين يتم نشرها في مواقع بعيدة أو معادية من أجل الكشف عن الأحداث. ونتيجة لذلك، يعد إدراج أنظمة حصاد الطاقة حلاً واعداً لإطالة عمر المستشعر وتحسين الأداء. في هذه الرسالة، نحن مهتمون بإستخدام تقنية حصاد الطاقة لشبكات الاستشعار اللاسلكية. وبشكل أكثر تحديداً، فإن الهدف هو كيفية تكييف تشغيل الشبكة مع دورات التفريغ والشحن الخاصة بالمستشعرات. وبشكل عام فهذا التكييف ماهو إلا طريقة الإنتقال بين حالتين (النوم والاستيقاظ) بناءً على تقنية دورة العمل (أي أن المستشعر ذو مستوى طاقة منخفض يختار فترة نوم أطول لتجميع أكبر قدر ممكن من الطاقة قبل الذهاب إلى حالة التفريغ (الاستيقاظ) حيث يتم استهلاك الطاقة المودعة). في الواقع، بما أن وقت إستيقاظ المستشعر لا يمكن تقديره بدقة، لأن معدل الطاقة الدقيق الذي يجمعه المستشعر يتغير بمرور الوقت، فمن الصعب للغاية ضمان توجيه الحزم. بالإضافة إلى ذلك، فإن عدم التيقن من الوقت الذي تستغرقه العقدة لحصاد ما يكفي من الطاقة قبل أن تتمكن من العمل مرة أخرى يعني أن حلول جدولة فترات النوم والاستيقاظ لجهاز استشعار المقترحة من قبل غير قابلة للإستخدام. من خلال إدخال سياسة عتبة الطاقة وإستخدام الطاقة المتبقية في بطارية جهاز الاستشعار، يتم تنظيم فترات اليقظة والنوم لتقليل دورة التشغيل لكل جهاز استشعار في الشبكة مع ضمان التوازن بين استهلاك الطاقة وقدرة حصاد الطاقة. تستخدم سياسة الحد الأدنى المقترحة للطاقة مرحلتين (أولهما، مرحلة التبديل بين ثلاث حالات محتملة لكل جهاز استشعار لتنظيم فترة الاستيقاظ وضمان توازن الحمل في الشبكة والثانية، تمثل في حساب مدة فترة النوم لتقليل من دورة العمل وجعلها ديناميكية). النتائج الرئيسية المحصل عليها من خلال إدراج هذه السياسة في الاتصالات المختلفة المستعملة بين أجهزة الاستشعار في هذا النوع من الشبكات، هي القدرة على تقليل من: عدد حالات التصادم، الحد الأقصى لعدد عمليات إعادة الإرسال، والتنافس على القناة في الشبكة. ونتيجة لذلك، سمح لنا إستخدام هذه السياسة بتمديد عمر الشبكة وتحسين أدائها. وبخلاصة لذلك، فإن زيادة عمر شبكة الإستشعار اللاسلكية هي مهمة معقدة وتستهلك الكثير من الوقت. وعلى الرغم من أن الخبرة المكتسبة في تطبيق تحسين إدارة الطاقة في شبكات أجهزة الاستشعار اللاسلكية لحصاد الطاقة مفيدة بشكل واضح في التغلب على بعض أوجه القصور في العديد من الأعمال القائمة وجعل هذه الإدارة أساسية وأكثر عملية.

الكلمات المفتاحية: شبكة الإستشعار اللاسلكية؛ حصاد الطاقة؛ دورة العمل؛ عمر شبكة الإستشعار اللاسلكية.

Résumé

La récupération d'énergie est le processus qui consiste à obtenir de l'énergie à partir de l'environnement externe (e.g., lumière solaire et artificielle, mouvements vibratoires, chaleur, etc.) et de transformer cette énergie en une énergie électrique, directement utilisable ou emmagasinée afin de servir au fonctionnement d'appareils de petite taille. Cette récolte de l'énergie ambiante peut être utilisée pour alimenter de petits capteurs autonomes sans fil qui ont de petites batteries et qui sont déployés dans des endroits éloignés ou hostiles pour la détection d'événements. De ce fait, l'intégration des systèmes de la récolte énergétique est une solution prometteuse pour prolonger la durée de vie des capteurs et améliorer ses performances. Dans cette thèse, nous nous intéressons à l'utilisation de la technique de récupération énergétique pour les réseaux de capteurs sans fil. Plus précisément, l'objectif est d'adapter le fonctionnement du réseau aux cycles de recharge/décharge des capteurs. Il s'agit d'une façon générale, de basculer entre deux états (veille et réveil) en se basant sur la technique duty-cycle (i.e., un capteur avec un niveau d'énergie inférieur sélectionne une période de veille plus longue pour collecter autant d'énergie que possible avant de passer à l'état de déchargement (réveil) où l'énergie déposée est consommée). En effet, puisque le réveil d'un capteur ne peut pas être estimé avec précision, du fait que le taux exact d'énergie récolté par le capteur fluctue dans le temps, il est très difficile de s'assurer de l'acheminement des paquets. De plus, l'incertitude sur le temps qu'il faut à un nœud afin de récolter assez d'énergie avant qu'il puisse fonctionner à nouveau fait en sorte que les solutions de planification de veille/réveil d'un capteur proposées dans la littérature sont inutilisables. Par l'introduction d'une politique de seuil d'énergie et l'utilisation de l'énergie restante au niveau de la batterie d'un capteur, les périodes de réveil et de veille sont réglementées afin de diminuer le duty-cycle de chaque capteur dans le réseau tout en assurant un équilibre entre la consommation d'énergie et la capacité de récupération d'énergie. La politique de seuil d'énergie proposée comprend deux phases (premièrement, la phase de commutation entre trois états possibles pour chaque capteur pour réguler sa période de réveil et d'assurer un équilibre de charge dans le réseau et la seconde, est le calcul de la durée de la période de veille afin d'optimiser le duty-cycle et le rendre dynamique). Les principaux résultats obtenus à l'aide de l'intégration de cette politique dans les différents types de communications utilisées entre les capteurs, sont la possibilité de minimiser le nombre de collisions, le nombre maximal de retransmissions et la contention sur le canal dans le réseau. De ce fait, l'utilisation de cette politique nous a permis de prolonger la durée de vie du réseau et d'améliorer ses performances.

Mots clés: Réseau de capteurs sans fil; La récupération d'énergie; duty-cycle; La durée de vie du réseau de capteurs sans fil.

Abstract

The energy harvesting is the process of obtaining energy from the external environment (e.g., solar and artificial light, vibratory motion, heat, etc.) and transforming this energy into electrical energy, directly usable or stored for use in the operation of small devices. This harvest of ambient energy can be used to power the small wireless autonomous sensors that have small batteries and are deployed in remote or hostile locations for event detection. Therefore, the integration of energy harvesting systems is a promising solution to extend the sensor lifetime and improve its performances. In this thesis, we are interested in the use of the energy harvesting technique for wireless sensor networks. More specifically, the objective is to adapt the operation of the network to the recharging/discharging cycles of the sensors. It's about a general way of switching between two states (sleep and wake-up) based on the duty-cycle technique (i.e., a sensor with a lower energy level selects a longer sleep period to harvest as much energy as possible before going to the unloading state (wake-up) where the deposited energy is consumed). Indeed, since the wake-up of a sensor cannot be accurately estimated, the fact that the exact rate of harvested energy by the sensor fluctuates over time, it is very difficult to ensure the routing of the packets. In addition, the uncertainty over the time it takes for a node to harvest enough energy before it can work again means that a sensor's sleep/wake-up planning solutions are proposed in the literature are unusable. By introducing an energy threshold policy and using the remaining energy in the battery of a sensor, the wake-up and sleep periods are regulated to decrease the duty cycle of each sensor in the network while ensuring a balance between energy consumption and the energy harvesting capacity. The proposed energy threshold policy comprises two phases (firstly, the switching phase between three possible states for each sensor to regulate its wake-up period and ensure load balance in the network and the second, is the calculation of the duration of the sleep period to optimize the duty cycle and make it dynamic). The main results achieved through the integration of this policy in the different types of communications used between sensors, are the possibility of minimizing the number of collisions, the maximum number of retransmissions, and contention for the channel in the network. Thereby, the use of this policy has allowed us to extend the network lifetime and improve its performance.

Keywords: Wireless sensor network; Energy harvesting; duty-cycle; Wireless sensor network lifetime.