

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي و البحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

Université Dr. Tahar MoulaySAIDA

جامعة د. الطاهر مولاي سعيدة

Faculté : Technologie

كلية : التكنولوجيا

Département : Informatique

قسم : الإعلام الآلي



MEMOIRE DE MASTER

Option :

Réseaux informatiques et systèmes répartis

THEME

Implémentation du protocole de routage T-MAC dans Simulateur OMNeT++.

Présenté par :

- Medjdoubi Mahmoud Yassine

Encadré par :

- Mr.Mekkaoui

Promotion : 2015 - 2016

Remerciements

Au terme de ce travail je tiens tout d'abord à remercier Allah le tout puissant de m'avoir donné la foi, la volonté et la persévérance pour l'aboutissement de ce travail.

Je tiens à exprimer toute ma reconnaissance à mon Directeur de mémoire **Mr Mekkaoui**. Je le remercie de m'avoir encadré, orienté, aidé et conseillé.

Nous tenons à remercier également les membres du Jury pour leur précieux temps accordé à l'étude de ce mémoire.

Nos remerciements et gratitude vont aux professeurs et enseignants, étudiants et personnel de notre l'université qui forment une famille très unie.

Dédicace

À mon père, mère, sœurs, frères, amis.

Liste des figures	04
Liste des tableaux	06
Introduction général	07
1. Les Réseaux de Capteurs Sans Fil	09
1.1 Introduction	09
1.2 Présentation des réseaux de capteurs	09
1.2.1 Définition d'un capteur	09
1.2.2 Architecture d'un nœud-capteur	10
1.2.3 Définition d'un RCSF ou WSN	11
1.2.4 Architecture d'un RCSF	12
1.3 Architecture protocolaire	13
1.4 Topologies de réseau de capteurs	14
1.5 Facteurs et contraintes de conception d'un RCSF	15
1.5.1 Contraintes conceptuelles	16
1.5.2 Contraintes matérielles	17
1.6 Applications des Réseaux de capteurs Sans Fil	17
1.7 Consommation d'énergie dans les RCSF	19
1.8 Conclusion	20
2. Le routage dans RCSFs	21
2.1 Introduction	21
2.2 Contraintes de routage dans les RCSFs	21
2.3 LES SOURCES DE GASPILLAGE D'ENERGIE	22
2.4 LES PROTOCOLES D'ECONOMIE ENERGIE	23
2.4.1 Protocoles dédiés à la couche réseau	23
2.4.2 Protocoles dédiés à la couche liaison (sous couche MAC)	26
2.5 Conclusion	30
3. Le protocole TMAC	31
3.1 Introduction	31
3.2 T-MAC PROTOCOL DESIGN	32
3.3 Basic Protocol	33
3.4 Clustering and synchronisation	34

3.5 Détermination de TA	35
3.6 Overhearing Avoidance	36
3.7 L'évitement du phénomène d'Overhearing	37
3.8 Le problème du « early-sleeping »	37
3.9 Communication asymétrique	38
3.9.1 FRTS (Future Request-to-send)	39
3.9.2 Taking priority on full buffers	40
3.10 Les avantages et Les inconvénients de TMAC	41
3.11 Conclusion	42
4. La simulation en RCSF	43
4.1 Introduction	43
4.2 Généralités sur la simulation	43
4.3 Types de simulation	44
4.4 Les simulateurs de réseau existant	44
4.4.1 NS2	44
4.4.2 GloMoSim	45
4.4.3 OMNET++	46
4.5 Comparaison entre les simulateurs	46
4.6 Le Simulateur OMNET++	47
4.6.1 Choix du simulateur OMNET++	48
4.6.2 Description architecturale d'OMNET++	48
4.6.3 Installation du simulateur OMNET++	49
4.6.4 Les principaux fichiers d'OMNET++	50
4.7 Les plates formes d'OMNET++	50
4.7.1 Mixim	51
4.7.2 Castalia	51
4.8 Détails sur Castalia	52
4.9 Outils de Simulation	54
4.9.1 Installation d'OMNET++ Version 4.6	54
4.9.2 Installation de Castalia3.2	54
4.10 Implémentation d'un réseau	55
4.11 Conclusion	55

5. Implémentation	56
5.1 Introduction	56
5.2 Environnement de travail	56
5.3 Réalisation	56
5.3.1 Objectif	56
5.4 Description détaillé	57
5.5 Résultats et analyses de simulation de T-MAC	59
5.6 Conclusion	63
 Conclusion générale	 64
Liste des abréviations	65
Bibliographie	67

Figure 1.1 : Architecture d'un nœud capteur.

Figure 1.2 : Architecture d'un RCSF.

Figure 1.3 : La pile protocolaire des réseaux de capteurs.

Figure 1.4 : Topologies des réseaux de capteurs.

Figure 1.5 : Quelques domaines d'applications des réseaux de capteurs sans fil.

Figure 2.1 : Fonctionnement du protocole LEACH.

Figure 2.2 : le mécanisme du Duty-cycling.

Figure 2.3 : La période d'activité et de mise en veille dans S-MAC.

Figure 2.4 : Sommeil et réveil cycles SMAC et TMAC.

Figure 2.5 : Protocole B-MAC.

Figure 3.1 : TMAC Protocol.

Figure 3.2 : Le schéma de protocole base T-MAC, avec adaptation actives moments.

Figure 3.3 : Choix du paramètre TA dans le protocole Timeout-MAC.

Figure 3.4 : Le problème du « early-sleeping ».

Figure 3.5 : L'échange de paquet Future Request To Send (FRTS).

Figure 3.6 : Prise de la priorité.

Figure 4.1 : Description architecturale du simulateur NS2.

Figure 4.2 : Transfert des paquets dans GloMoSim.

Figure 4.3 : Le lancement du simulateur Omnet++.

Figure 4.4 : Architecture modulaire du simulateur Omnet++.

Figure 4.5 : Exécution d'une simulation sous OMNeT++.

Figure 4.6 : Les connections des modules sous Castalia.

Figure 4.7 : Les principaux composants d'un nœud.

Figure 4.8 : Les étapes Installation d'OMNET++ Version 4.6.

Figure 5.1 : Fichier graphique ".ned" du nœud.

Figure 5.2 : Le fichier SensorNetwork.ned.

Figure 5.3 : Fichier résultat.txt du Castalia.

Figure 5.4 : Fichier trace.txt du Castalia.

Figure 5.5 : L'énergie consommée par chaque nœud.

Figure 5.6 : détail des paquets reçus.

Figure 5.7 : L'énergie moyenne consommée par plusieurs simulations.

Figure 5.8 : Nombre de paquets transmis par plusieurs simulations.

Tableau 4.1 : les avantages et inconvénients des simulateurs décrit auparavant.

Tableau 5.1 : Les paramètres utilisés pour la simulation de protocole TMAC.

Introduction générale

Les réseaux de capteurs sans fil ont un aspect supplémentaire: comme nœuds de capteurs sont en général fonctionnant sur batterie, la consommation d'énergie c'est très important. L'interface radio est l'équipement le plus consommateur en énergie dans un nœud capteur. Comme la couche MAC contrôle l'accès au médium c'est-à-dire l'utilisation de cette interface pour transmettre les messages, c'est le protocole MAC qui est le plus concerné par la consommation énergétique dans un réseau de capteurs sans fil. Plusieurs recherches ont été conduites afin de proposer des protocoles MAC avec une gestion efficace d'énergie.

La communication dans les réseaux de capteurs sans fil peut, comme la plupart la communication de réseau, être divisée en plusieurs couches. Une de celles-ci est la couche de contrôle d'accès au support (MAC). Cette couche est décrite par un protocole MAC, qui cherche à s'assurer que deux nœuds interfèrent avec les transmissions de l'autre, et traite de la situation quand ils le font.

Le routage dans les réseaux de capteurs sans fil est la procédure d'acheminement d'informations d'un nœud source collecteur vers une destination à travers un réseau de connexion. Chaque nœud est susceptible d'être à la disposition des autres nœuds capteurs pour participer à la transmission et retransmission des informations émises sur le réseau par un ou un ensemble de nœuds n'ayant pas la possibilité d'atteindre directement la destination.

Les protocoles traditionnels MAC sont conçus pour maximiser le débit, de minimiser la latence et fournir l'équité, la conception de protocoles pour les réseaux de capteurs sans fil se concentre sur la réduction de la consommation d'énergie. L'application détermine les exigences relatives à la (modeste) le débit minimum et une latence maximale. L'équité est généralement pas un problème, étant donné que les nœuds dans un réseau de capteurs sans fil sont typiquement partie d'une seule application et de travailler ensemble pour un commun objectif.

Le protocole T-MAC (Timeout MAC), une énergie adaptative efficace protocole MAC pour les réseaux de capteurs sans fil. qui minimise l'écoute au ralenti, tout en considérant de capteurs sans fil des schémas de communication et des limitations matérielles, des solutions d'économie d'énergie existantes.

T-MAC, un moyen basé sur la contention Contrôle d'accès protocole MAC pour les réseaux de capteurs sans fil. Les demandes de ces réseaux ont des caractéristiques (Faible débit de messages, l'insensibilité à la latence) qui peut être exploitée pour réduire la consommation d'énergie en introduisant un cycle de service actif / sommeil. Pour gérer les variations de charge dans le temps et l'emplacement T-MAC présente un cycle de service adaptatif une nouvelle façon: en mettant fin dynamiquement la partie active de celui-ci. Cela réduit la quantité d'énergie gaspillée sur l'écoute de repos, dans lequel les nœuds attente pour les messages entrants potentiellement, tandis que en conservant un débit raisonnable.

Dans notre travaille on s'intéresse à l'implémentation de l'un des protocoles MAC proposé par les chercheurs, le protocole (T-MAC) Time-Out Médium Access Control, on utilise le simulateur OMNeT++ avec la plate forme « Castalia » afin de faire une étude sur les performances et l'efficacité de ce protocole en terme d'économie d'énergie.

Ce mémoire est organisé en cinq chapitres :

1. Dans le premier chapitre, nous présentons des généralités sur les réseaux de capteurs sans fil.
2. Le deuxième chapitre est présenter la notion de routage et classifiez les protocoles mac dans les réseaux de capteurs sans fil.
3. Le troisième chapitre est une présentation détaillé de fonctionnement du protocole TMAC.
4. Le quatrième chapitre présente les outils logiciels nécessaires à la réalisation de notre travail.
5. Le cinquième chapitre constitue le cœur de notre travail.
6. On termine par une conclusion récapitulative.

1.1. Introduction :

Les progrès dans le domaine de l'électronique miniaturisée et les communications sans fil ont donné naissance à des composants capables de prélever des grandeurs environnementales, physiologiques etc. Ces composants sont appelés des nœuds capteurs et ils ont la capacité de s'auto-organiser pour former un réseau de capteurs sans fil (RCSF).

Les RCSF permettent de faciliter le suivi et le contrôle à distance de l'environnement physique avec une meilleure précision. Ils peuvent aussi être déployés pour exploiter diverses applications (environnementales, militaires, médicales, etc.). En outre, un réseau de capteurs est constitué généralement d'un grand nombre de nœuds capteurs car ces derniers sont sujets à pannes accidentelles ou intentionnelles. Chaque nœud est composé principalement d'un ou plusieurs capteurs, d'une unité de traitement et d'un module de communication. Ces nœuds communiquent entre eux selon une certaine topologie du réseau afin d'acheminer les informations à un centre de contrôle distant de la zone de leur déploiement. La mise en place d'un RCSF pose de nombreux problèmes parmi lesquels le routage des informations vers la station de base via les différents nœuds du réseau. Dans cette optique plusieurs contributions ont été proposées dans la littérature. Ces contributions visent à minimiser la consommation d'énergie ceci afin d'optimiser l'autonomie des nœuds qui constituent le réseau et par suite garantir une longue longévité pour le réseau entier [01].

1.2. Présentation des réseaux de capteurs

1.2.1. Définition d'un capteur

Les capteurs sont des dispositifs capables de générer des données relatives à leur environnement physique. Ces dispositifs embarquent un système de communications afin d'échanger des données formant ainsi un réseau implicite.

1.2.2. Architecture d'un nœud capteur

Un capteur est composé principalement d'une unité de : captage, traitement, stockage, communication, et énergie. Des composants additionnels peuvent être ajoutés selon le domaine d'application [02]. Comme par exemple un système de localisation tels qu'un GPS (Global Positioning System), un générateur d'énergie (exemple : cellules solaires) ou un mobilisateur lui permettant de se déplacer. Ces éléments principaux et optionnels (représentés par les traits discontinus) sont visibles sur la figure I.1.

- 1. Unité d'énergie :** Un capteur est muni d'une source d'énergie, généralement une batterie, pour alimenter tous ses composants. Les batteries utilisées sont soit rechargeables ou non. Souvent, dans les environnements sensibles, il est impossible de recharger ou changer une batterie. Pour cela, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs et donc d'un réseau de capteurs.
- 2. Unité de captage :** La fonction principale de l'unité de captage est de capturer ou mesurer les données physiques à partir de l'objet cible. Il est composé de 2 sous-unités : le récepteur (reconnaissant la grandeur physique à capter) et le transducteur (convertissant le signal du récepteur en signal électrique). Le capteur fournit des signaux analogiques, basés sur le phénomène observé, au Convertisseur Analogique/Numérique (CAN). Ce dernier transforme ces signaux en données numériques et les transmet à l'unité de traitement. Un capteur peut avoir un ou plusieurs unités de captage [03].

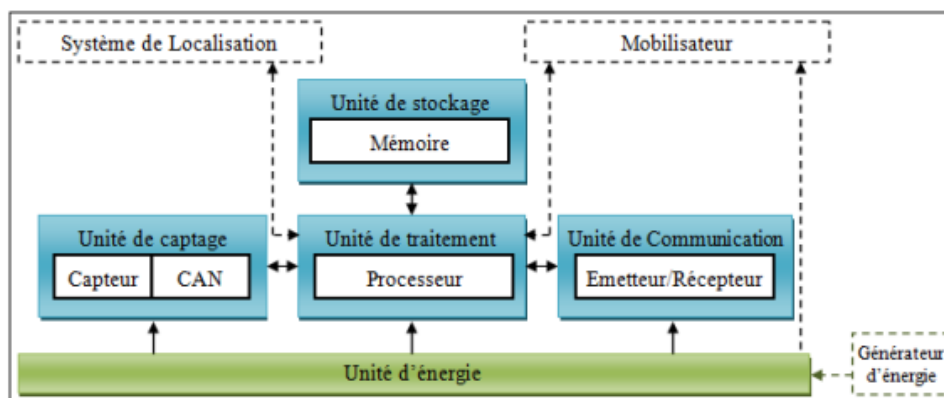


Figure 1.1 : Architecture d'un nœud capteur.

- 3. Unité de traitements (processeur) :** Elle est composée d'une interface d'acquisition et d'une interface pour l'unité de transmission ainsi qu'un processeur et un système d'exploitation spécifique. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de transmission. Les types de processeurs qui peuvent être utilisés dans un capteur incluent le Microcontrôleur, les DSP (Digital Signal Processors), les FPGA (Field Programmable Gate Array) et les ASIC (Application Specific Integrated Circuit). Parmi toutes ces alternatives, le Microcontrôleur a été le processeur le plus utilisé pour les capteurs à cause de sa flexibilité à être relié à d'autres composants (comme par exemple l'unité de communication), à son bon prix et sa faible consommation énergétique.
- 4. Unité de communication :** Elle permet l'échange de données via un support de communication sans fil. Les différents choix de média de transmission incluent la Radio Fréquence (RF), le Laser et l'infrarouge. Cette unité représente la partie matérielle la plus consommatrice d'énergie [04].
- 5. Unité de stockage(Mémoire) :** Elle inclut la mémoire de programme et la mémoire de données. La taille de cette mémoire est souvent limitée essentiellement par les considérations économiques et elle est en continuelle amélioration au fil des années.

1.2.3. Définition d'un RCSF ou WSN

Un réseau de capteurs sans fil est un type spécial de réseaux ad hoc avec un grand nombre de nœuds qui sont des micro-capteurs capables de recevoir et de transmettre des données environnementales d'une manière autonome sans intervention humaine.

La position de ces nœuds n'est pas obligatoirement prédéterminée, ils peuvent être aléatoirement dispersés dans une zone géographique appelée « champ de captage » correspondant au terrain d'intérêt pour le phénomène capté (par exemple : lâchée de capteurs sur un volcan pour étudier les phénomènes volcanologiques et leurs évolutions) [04]. Le réseau possède en général un nœud particulier, la base (ou sink), connectée avec les autres nœuds par un réseau filaire est reliée à une alimentation électrique.

1.2.4. Architecture d'un réseau de capteurs sans fil

Un réseau de capteur sans fil(RCSF), consiste en un ensemble de nœuds capteurs variant de quelques dizaines d'éléments à plusieurs milliers, placés de manière plus ou moins aléatoire (par exemple par largage depuis un hélicoptère), dans une zone géographique appelée zone de captage, ou zone d'intérêt, afin de surveiller un phénomène physique et de récolter leurs données d'une manière autonome.

Les nœuds capteurs utilisent une communication sans fil pour acheminer les données captées vers un nœud collecteur appelé nœud puits (sink en anglais), ou station de base (base station). Le puits transmet ensuite ces données par Internet ou par satellite à l'ordinateur Central « Gestionnaire de tâches » pour analyser ces données et prendre des décisions. Ainsi, l'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises, puis récolter les données environnementales captées par le biais du nœud puits.

En plus des nœuds capteurs, le modèle peut introduire les super-nœuds, appelés des passerelles (Gateways) [04]. Ces derniers possèdent une source d'énergie importante, la capacité de traitement et de stockage plus élevées comparativement aux nœuds capteurs. Ils peuvent ainsi être utilisés pour exécuter les tâches plus complexes comme la fusion des données issues des capteurs d'une même zone.

Dans le cas le plus simple, les nœuds capteurs seront dans le voisinage direct du puits (communication à un à un saut). Cependant, dans le cas d'un réseau à grande échelle, ils ne sont pas tous dans le voisinage du puits et les données seront acheminées du nœud source vers le puits en transitant par plusieurs nœuds, selon un mode de communication multi-sauts comme l'illustre la figure I.2.

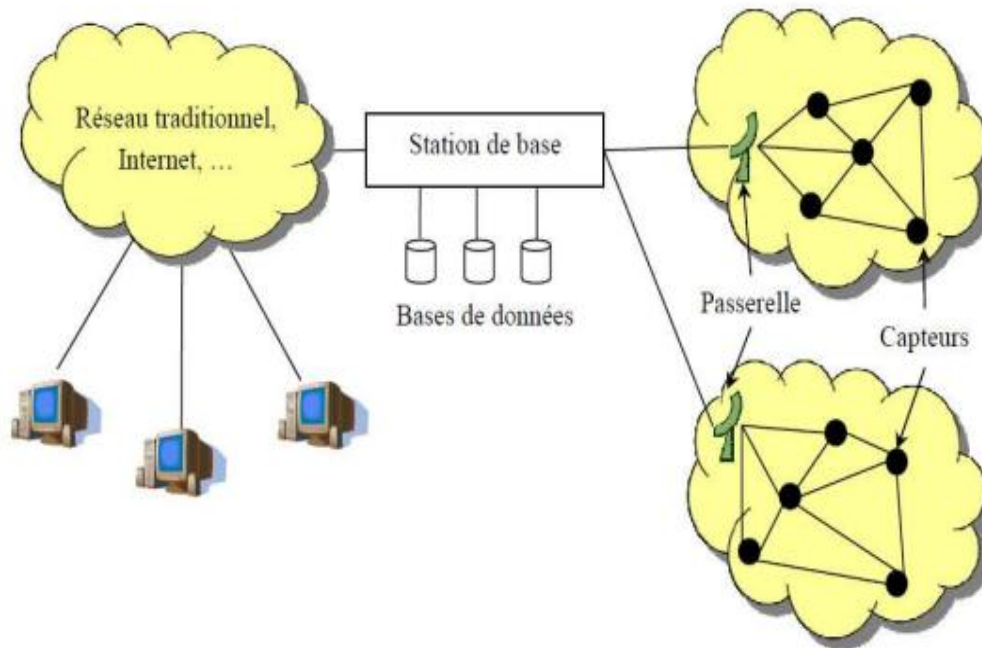


Figure 1.2 : Architecture d'un RCSF.

1.3. Architecture protocolaire

La pile de protocoles utilisée par le puits (Sink) ainsi que par tous les nœuds-capteurs est donnée dans la Fig. I.3. Cette pile de protocoles combine routage et gestion d'énergie et intègre les données avec les protocoles réseau [06]. Elle communique de manière efficace (en termes d'énergie) à travers le support sans fil et favorise les efforts de coopération entre les nœuds-capteurs.

La pile de protocoles comprend une couche application, une couche transport, une couche réseau, une couche liaison de données, une couche physique, un plan de gestion d'énergie, un plan de gestion de mobilité et un plan de gestion des tâches. Selon les tâches de détection, différents types de logiciels d'application peuvent être construits et utilisés dans la couche application. La couche transport contribue au maintien du flux de données si l'application du réseau de capteurs l'exige. La couche réseau s'occupe de l'acheminement des données fournies par la couche transport.

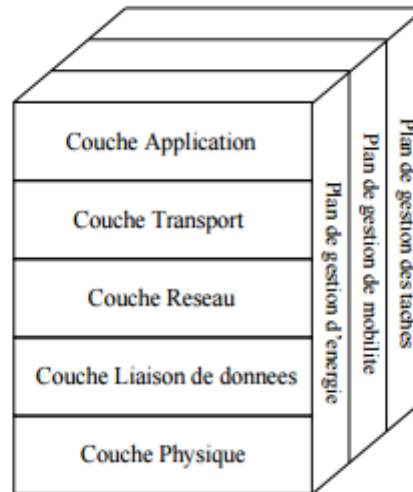


Figure 1.3 : La pile protocolaire des réseaux de capteurs.

Comme l'environnement est sujet au bruit et que les nœuds-capteurs peuvent être mobiles, le protocole MAC doit tenir compte de la consommation d'énergie et doit être en mesure de réduire les collisions entre les nœuds voisins lors d'une diffusion par exemple [05]. La couche physique répond aux besoins d'une modulation simple mais robuste, et de techniques de transmission et de réception. En outre, les plans de gestion d'énergie, de mobilité et des tâches surveillent et gèrent la consommation d'énergie, les mouvements, et la répartition des tâches entre les nœuds-capteurs. Ces plans aident les nœuds-capteurs à coordonner les tâches de détection et à réduire l'ensemble de la consommation d'énergie.

1.4. Topologies des réseaux de capteurs

Un réseau de capteurs sans fil est composé d'un ensemble de nœuds capteurs et des Gateway qui s'occupent de collecter les données des capteurs et de les transmettre à l'utilisateur via l'internet ou le satellite [07], il existe plusieurs topologies pour les réseaux de capteurs :

- **Topologie en étoile** : est un système uni-saut. Tous les nœuds envoient et reçoivent seulement des données avec la station de base. Cette topologie est simple et elle demande une faible consommation d'énergie, mais la station de base est vulnérable et la distance entre les nœuds et la station est limitée.

- **Topologie en toile (Mesh Network)** La topologie en toile est un système multi-saut. La communication entre les nœuds et la station de base est possible. Chaque nœud a plusieurs chemins pour envoyer les données.
Cette topologie a plus de possibilités de passer à l'échelle du réseau, avec redondance et tolérance aux fautes, mais elle demande une consommation d'énergie plus importante.
- **Topologie hybride :** La topologie hybride est un mélange des deux topologies ci-dessus. Les stations de base forment une topologie en toile et les nœuds autour d'elles sont en topologie étoile. Elle assure la minimisation d'énergie dans les réseaux de capteurs.

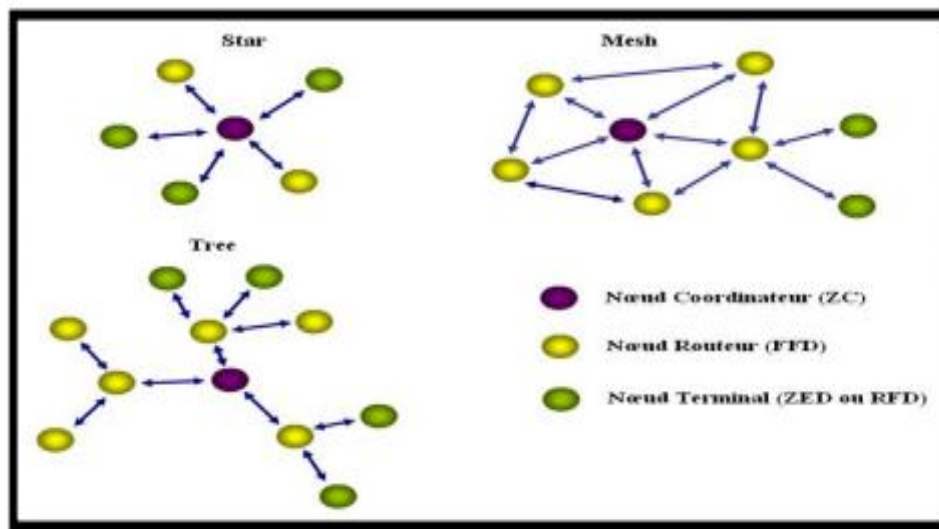


Figure 1.4 : Topologies des réseaux de capteurs.

1.5. Facteurs et contraintes de conception d'un RCSF

La conception et la mise en place des RCSFs sont influencées par plusieurs contraintes qui peuvent être des contraintes conceptuelles ou matérielles [08]. Ces facteurs importants servent comme directives pour le développement des algorithmes et protocoles utilisés dans les réseaux de capteurs [09] ; ils sont considérés également comme métriques de comparaison de performances entre les différents travaux dans le domaine.

1.5.1. Contraintes conceptuelles

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

- ❖ **La tolérance aux pannes :** La défaillance ou le blocage de certains nœuds dans un réseau de capteurs peut être engendrés par plusieurs causes, notamment l'épuisement d'énergie, l'endommagement physique, ou les interférences liées à l'environnement. Ces problèmes ne devraient pas affecter le reste du réseau. C'est le principe de la tolérance aux pannes.
- ❖ **L'extensibilité (passage à l'échelle) :** L'une des caractéristiques des RCSF est qu'ils peuvent contenir des centaines voire des milliers de nœuds capteurs. Suivant l'application, ce nombre peut encore augmenter jusqu'à des millions de capteurs. Les nouveaux schémas doivent pouvoir garantir un bon fonctionnement avec ce nombre élevé de capteurs. Ils doivent aussi exploiter la nature fortement dense des réseaux de capteurs.
- ❖ **Le Coût de production :** Le coût de production d'un seul micro-capteur est très important pour l'évaluation du coût global du réseau [09], si ce dernier est supérieur à celui nécessaire pour le déploiement des capteurs classiques, l'utilisation de cette nouvelle technologie ne serait pas rentable.
- ❖ **Environnement :** Les nœuds capteurs doivent être conçus d'une manière à résister aux différentes et sévères conditions de l'environnement : forte chaleur, pluie, humidité...
- ❖ **Media de transmission :** Les nœuds communicants sont reliés sans fil. Ce lien peut être réalisé par radio, signal infrarouge ou un média optique.
- ❖ **Consommation énergétique :** Les nœuds capteurs, étant des dispositifs microélectroniques, peuvent être équipés seulement d'une source d'énergie limitée (<0.5 Ah, 1.2 V). Dans certains scénarios d'application, il est impossible de réapprovisionner de l'énergie. La durée de vie d'un capteur est donc dépendante de la durée de vie de sa batterie. D'autre part, la retransmission des données, la réorganisation du réseau ainsi que le changement de sa topologie rendent la gestion et la conservation d'énergie d'une haute importance.

Cette énergie est consommée par les différentes unités du capteur afin de réaliser les tâches de captage, traitement de données et communication. Cette dernière est l'opération qui consomme le plus d'énergie.

- ❖ **Agrégation de données :** Dans les RCSF, les données produites par les nœuds capteurs voisins sont très corrélées spatialement et temporellement. Ceci peut engendrer la réception par la station de base d'informations redondantes.

Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau et ainsi d'améliorer sa durée de vie. L'une des techniques utilisée pour réduire la transmission d'informations redondantes est l'agrégation des données, appelée aussi fusion de données.

1.5.2. Contraintes matérielles.

Parmi les contraintes matérielles liées aux RCSFs [09], on peut citer :

- ❖ **Dimension :** La taille réduite des capteurs peut présenter de nombreux avantages, elle permet un déploiement flexible et simple du réseau. Cependant, la puissance des batteries utilisées pour alimenter les nœuds capteurs est limitée, par la petite taille de ces derniers.
- ❖ **Puissance de calcul :** Les processeurs des réseaux de capteurs sont différents de ceux d'une machine classique. Car ils utilisent souvent des microcontrôleurs de faibles fréquences.

1.6. Applications des Réseaux de capteurs Sans Fil

Les réseaux de capteurs sans fil (RCSF) ont un champ d'application vaste et diversifié. Ceci est rendu possible par leur coût faible, leur taille réduite, le support de communication sans fil utilisé et la large gamme des types de capteurs disponibles. Un autre avantage est la possibilité de s'auto-organiser et d'établir des communications entre eux sans aucune intervention humaine, notamment dans des zones inaccessibles ou hostiles, ce qui accroît davantage le nombre de domaines ciblés par leur application (environnement, catastrophes naturelles, bâtiments intelligents, la santé, l'agriculture, l'industrie...etc.). Nous présentons dans ce qui suit les domaines les plus ciblés par les RCSF [10] :

- ✓ **Domaine militaire :** Les RCSF sont le résultat de la recherche militaire. Ils sont utilisés dans la surveillance des champs de bataille pour connaître exactement la position, le nombre, l'armement (chimique, biologique, nucléaire...etc.), l'identité et le mouvement des soldats et ainsi empêcher leur déploiement sur des zones à risques.
Exemple d'application : Le déploiement d'un réseau de capteurs sans fil sur les côtes Est et Ouest des Etats-Unis destiné à repérer les sous-marins silencieux et les navires, notamment soviétiques, s'approchant du territoire américain.
- ✓ **Domaine civil :** Apparus dans plusieurs contextes notamment dans la surveillance des habitations (concept de bâtiments intelligents), des infrastructures, des installations et des zones à risques. Leur utilisation permet de réduire considérablement le budget consacré à la sécurité des humains tout en garantissant des résultats sûrs et fiables.
- ✓ **Domaine agricole et environnemental :** Les réseaux de capteurs sans fil sont très utiles dans la protection de l'environnement. Ils peuvent être utilisés pour la détection des feux de forêts, des inondations, surveillance des volcans, contrôle de la qualité de l'air par le suivi de l'évolution de la densité moyenne de CO₂, le déplacement des animaux...etc. Dans le domaine agricole, on cite le déploiement des capteurs sur un champ agricole afin d'identifier les zones sèches et permettre leur irrigation à temps.
Exemple d'application : Le projet CORIE. On déploie des capteurs temps-réel pour récolter des informations visant à fournir des indications objectives sur la variabilité spatiale et temporelle de la partie inférieure de la rivière Columbia.
- ✓ **Domaine industriel :** Le suivi des chaînes de production dans une usine, détection des dysfonctionnements de machines, suivi du mouvement des marchandises dans les entrepôts de données, suivi du courrier, des colis expédiés...etc. sont, entre autres, des exemples concrets d'application des RCSF dans le domaine industriel.
- ✓ **Domaine de la santé :** Un moyen très efficace pour le domaine médical et le suivi temps-réel de l'état des patients, notamment ceux atteints de maladies chroniques, ils permettent de collecter des informations physiologiques de meilleure qualité pouvant être stockées pour une longue durée ou alors détecter des comportements anormaux chez des personnes âgées ou handicapées comme les chutes, les chocs, les cris...etc.
Exemple d'application : Utilisation de gélules multi-capteurs pour la transmission d'images de l'intérieur du corps humain sans avoir recours à la chirurgie.
- ✓ **Applications domestiques :** Les RCSF sont l'un des moyens les plus importants dans la lutte contre le réchauffement climatique. En effet, l'intégration des capteurs

dans des murs ou sur les plafonds des maisons permet d'économiser un maximum d'énergie en gérant au mieux l'éclairage et le chauffage en fonction de la localisation des personnes. Egalement, les capteurs peuvent être embarqués dans des appareils électroménagers (aspirateurs, micro-ondes, climatiseurs, réfrigérateurs...etc.) et d'interagir entre eux et avec un réseau externe pour assurer un meilleur contrôle à distance de ces appareils par leur propriétaire.

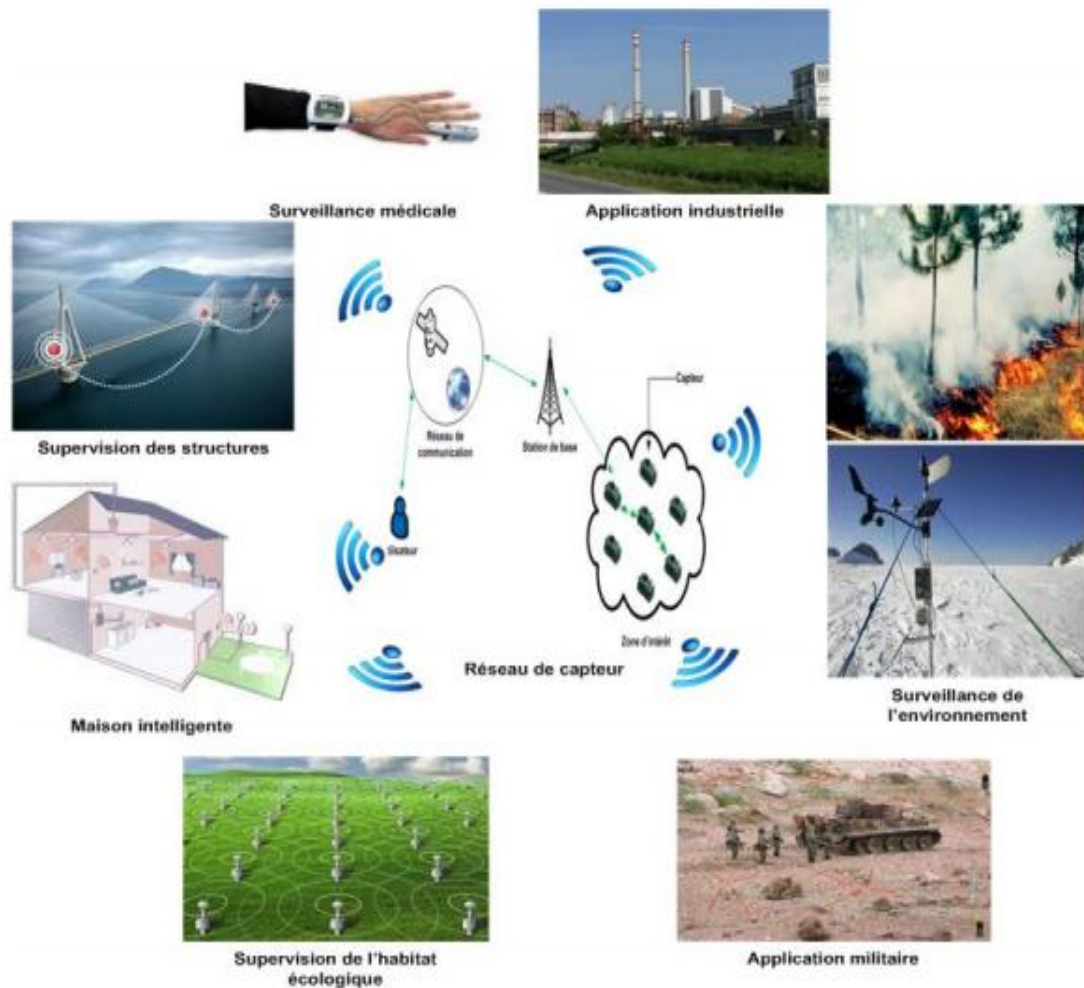


Figure 1.5 : Quelques domaines d'applications des réseaux de capteurs sans fil.

1.7. Consommation d'énergie dans les RCSF

L'énergie consommée par un nœud capteur est due essentiellement aux opérations suivantes : la capture, le traitement et la communication de données [11].

- **Energie de capture :** est dissipée pour accomplir les tâches suivantes : échantillonnage, traitement de signal, conversion analogique/numérique et activation de la sonde de capture. En général, l'énergie de capture représente un faible pourcentage de l'énergie totale consommée par un nœud.
- **Energie de traitement :** se divise en deux parties : l'énergie de commutation et l'énergie de fuite. L'énergie de commutation est déterminée par la tension d'alimentation et la Capacité totale commutée au niveau logiciel (en exécutant un logiciel). Par contre, l'énergie de fuite correspond à l'énergie consommée lorsque l'unité de calcul n'effectue aucun traitement. En général, l'énergie de traitement est faible par rapport à celle nécessaire pour la communication.
- **Energie de communication :** se décline en deux parties : l'énergie de réception et l'énergie de l'émission. Cette énergie est déterminée par la quantité des données à communiquer et la distance de transmission, ainsi que par les propriétés physiques du module radio. L'émission d'un signal est caractérisée par sa puissance. Quand la puissance d'émission est élevée, le signal aura une grande portée et l'énergie consommée sera plus élevée. Notons que l'énergie de communication représente la portion la plus grande de l'énergie consommée par un nœud capteur.

1.8. Conclusion

Dans ce chapitre, nous avons présenté les réseaux de capteurs, en parlant sur l'architecture, composants, fonctionnement, topologies utilisés, applications, les systèmes d'exploitation ainsi que les concepts généraux liés aux réseaux de capteurs sans fil. Parmi ces concepts nous avons identifié les contraintes de conception des réseaux de capteurs sans fil. Parmi ces contraintes nous avons vu les contraintes matérielles et conceptuelles.

2.1. Introduction :

Le routage dans les réseaux de capteurs sans fil est la procédure d'acheminement d'informations d'un nœud source collecteur vers une destination à travers un réseau de connexion. Chaque nœud est susceptible d'être à la disposition des autres nœuds capteurs pour participer à la transmission et retransmission des informations émises sur le réseau par un ou un ensemble de nœuds n'ayant pas la possibilité d'atteindre directement la destination.

En raison des caractéristiques particulières des RCSFs qui les distinguent des autres réseaux traditionnels sans fil, notamment en capacités de calculs, de stockage et d'énergie des nœuds capteurs, le routage est une tâche ardue et compliquée nécessitant un travail de collaboration de tous les nœuds appartenant au réseau.

Le problème qui se pose est l'adaptation de la méthode d'acheminement utilisée avec le grand nombre d'unités existant dans un environnement caractérisé par de modestes capacités de calcul et de sauvegarde et de changements rapides de topologies [12]. Il semble donc important que toute conception de protocole de routage doive étudier les problèmes suivants :

1. La minimisation de la charge du réseau.
2. Offrir un support pour pouvoir effectuer des communications multipoints fiables.
3. Assurer un routage optimal et offrir une bonne qualité concernant le temps de latence.

2.2. Contraintes de routage dans les RCSFs :

Le routage dans les réseaux de capteurs diffère de celui des réseaux Ad Hoc [13] dans les points suivants :

- ❖ il n'est pas possible d'établir un système d'adressage global pour le grand nombre de nœuds.
- ❖ les applications des réseaux de capteurs exigent l'écoulement de données mesurées depuis des sources multiples vers la destination finale « Sink ».
- ❖ les différents capteurs peuvent générer produire les mêmes données à proximité d'un phénomène (problème de la redondance des données).
- ❖ les nœuds capteurs exigent ainsi une gestion soignée des ressources.

2.3. LES SOURCES DE GASPILLAGE D'ÉNERGIE :

La surconsommation d'énergie est toute énergie consommée au-delà du seuil normal. Celle-ci peut être causée par différents [13] phénomènes qui engendrent le gaspillage d'énergie des nœuds capteurs.

- **L'écoute passive (idle) :** les nœuds capteurs sortent périodiquement de leur état de sommeil afin d'écouter le trafic dans le réseau. Cette écoute permet de savoir s'il y'a des données à recevoir ou à relayer dans le réseau. Cependant, cette mise en écoute peut être très coûteuse dans le cas de réseau à faible trafic. En effet, le nœud capteur peut écouter le trafic sans pour autant recevoir ou transmettre des données, ce qui va gaspiller inutilement ses réserves d'énergie. De plus, la transition périodique entre le sommeil et l'écoute engendrera une autre source de surconsommation d'énergie, surtout dans le cas où celle-ci n'est pas contrôlée.
- **Les collisions :** à cause de leur environnement de communication sans fil, les réseaux de capteurs sont fortement exposés aux interférences et aux collisions. Ces dernières sont générées lorsque deux ou plusieurs nœuds adjacents transmettent leurs données en même temps. En effet, les collisions sont considérées comme étant la source de gaspillage d'énergie la plus importante, étant donné que celles-ci provoquent la retransmission des paquets en collisions, ce qui est très coûteux en énergie.
- **La puissance de transmission :** la portée des antennes radio est directement liée à la puissance de transmission utilisée. La plupart des nœuds capteurs possèdent des antennes à portée statique dans lesquels la puissance de transmission est fixée précédemment par les concepteurs. Par conséquent, un nœud capteur peut gaspiller pas mal d'énergie en utilisant une grande puissance de transmission afin de communiquer avec un nœud très proche de lui.
- **Les distances de transmission :** afin d'économiser la consommation d'énergie, il est préférable de réduire les distances de transmission entre les nœuds capteurs. Ainsi, la communication multi sauts est souvent sollicitée dans les réseaux de capteurs sans fil, contrairement à celle basée sur un seul saut dans laquelle la dissipation d'énergie est très élevée.
- **L'écoute abusive (Overhearing) :** le nœud capteur peut recevoir toutes les données échangées entre ses nœuds voisins, même si ces dernières ne lui sont pas destinées. L'intensité de cette écoute abusive est proportionnelle à la densité du réseau.

Ainsi, cela peut engendrer un grand gaspillage d'énergie, vu que la majorité des RCSFs sont déployés à grande échelle.

- **Le surcoût des paquets de contrôle (Overhead):** l'échange des paquets de contrôle peut être une autre source de gaspillage d'énergie, principalement si le nombre de paquets de contrôle est inutilement élevé.

2.4. LES PROTOCOLES D'ECONOMIE ENERGIE:

L'économie d'énergie est autant considérée comme un paramètre fondamental dans le contexte de robustesse et de sécurité des réseaux de capteurs sans fil. Malgré les progrès qui ont été réalisés, le prolongement de la durée de vie des nœuds capteurs continue d'être un défi majeur et un facteur clé, exigeant davantage de recherches sur l'efficacité énergétique des plates-formes et des protocoles de communication [14]. Dans cette section, nous présentons un état de l'art sur les recherches proposées dans le domaine d'économie d'énergie dans les RCSFs. En effet, la plus grande partie ces recherches est réservée aux protocoles mono couche. Qui traitent le problème d'économie d'énergie au niveau d'une seule couche du modèle OSI.

2.4.1. Protocoles dédiés à la couche réseau :

Les protocoles de routage destinés pour la gestion efficace d'énergie, représentent la majorité des recherches proposées au niveau de la couche réseau. L'idée de base de cette catégorie de protocoles est la construction des chemins de routage à faible consommation énergétique (energy aware routing path). De nombreuses stratégies de routage ont été créées pour les réseaux de capteurs sans fil. Certaines sont des adaptations de stratégies qui existaient pour d'autres types de réseaux (principalement pour les réseaux sans fil au sens le plus large), tandis que d'autres ont été conçues spécialement pour les réseaux de capteurs sans fil. Les algorithmes de routage sont en fait découpés en trois familles [14] : les algorithmes de routage centrés données, hiérarchiques ou géographiques.

A. Les protocoles de routage centrés données (Data-centric protocols):

Le routage centré sur les données est le modèle le plus simple où chaque nœud dans le réseau transmet ses données à la station de base. Chaque nœud joue typiquement le même rôle et les nœuds capteurs collaborent pour accomplir la tâche de captage [15].

La station de base envoie des requêtes à certaines régions et se met en attente des données des capteurs situés dans les régions choisies. On peut citer comme exemples : les protocoles de propagation (flooding) et discussion (gossiping), le protocole de routage par négociation SPIN et le protocole de routage par diffusion dirigée.

Inondation : (“flooding”, en anglais) [16] consiste à envoyer un message à tout le réseau. L’émetteur envoie le message à tous ses voisins. Chaque voisin envoie à son tour le message à tous ses voisins et ainsi de suite. Les nœuds vont donc recevoir le même message plusieurs fois de différents voisins. Pour éviter que le message ne se multiplie dans le réseau, chaque nœud ne le renvoie qu’une seule fois. Pour ce faire, chaque message envoyé en inondation a un identifiant unique. Les nœuds qui réémettent le message notent l’identifiant. S’ils reçoivent à nouveau un message avec cet identifiant, ils ne le renvoient pas.

SPIN: L’idée derrière le SPIN [17] est de nommer les données en utilisant des descripteurs de haut niveau ou des métas données. Avant la transmission, les méta-données sont échangées entre les capteurs par un mécanisme de publicité de données. Chaque nœud recevant de nouvelles données, l’annonce à ses voisins et ceux intéressés récupèrent les données en envoyant une requête

B. Les protocoles de routage basés sur la localisation (géographique) :

Dans ce type de routage [14], les nœuds capteurs sont adressés en fonction de leurs localisations. La distance entre les nœuds voisins peut être estimée sur la base des forces entrantes du signal. Des coordonnées relatives des nœuds voisins peuvent être obtenues en échangeant une telle information entre les voisins. Alternativement, la location des nœuds peut être disponible directement en communiquant avec un satellite en utilisant GPS (système de positionnement global). Dans la plupart des protocoles de routage, l’information sur la localisation des nœuds est nécessaire afin de calculer la distance entre deux nœuds particuliers de sorte que la consommation d’énergie puisse être estimée. En effet, le routage géographique suppose que tous les nœuds connaissent leur position. Néanmoins, une solution basée sur le GPS peut être trop coûteuse, d’autant plus que le nombre de nœuds à équiper est très grand. Parmi ces protocoles géographiques, on peut mentionner les protocoles MECN , SMECN, GAF et GEAR.

GAF (Geographic Adaptive Fidelity): Proposé par Y. Xu et autres en 2001 [18], ce protocole découpe le réseau en plusieurs zones virtuelles; chaque nœud est affecté à une zone donnée selon ses coordonnées géographiques (en utilisant le GPS). Les nœuds appartenant à la même zone sont considérés comme équivalents en coût de routage; ainsi, on choisit seulement un nœud de chaque zone pour router les données et les autres sont mis en mode sommeil pour conserver de l'énergie. De cette façon, la durée de vie du réseau est augmentée en fonction du nombre total des nœuds inactifs. Dans GAF le nœud peut se retrouver dans l'un des trois cas possibles: Mode découvert: pour déterminer ses voisins de zone; Actif: s'il participe dans le routage de données; Endormi: s'il ne participe pas au routage.

Pour supporter la mobilité, chaque nœud estime et diffuse le temps nécessaire pour quitter sa zone à ses voisins. Sur la base de ce temps ils estiment leurs temps de réveil pour choisir celui qui va prendre le relais parmi eux. Il faut noter que GAF ne prend pas en compte l'énergie du nœud lors du choix du représentant, ce qui peut générer des trous dans le réseau une fois que ce dernier épuise son énergie.

C. Les protocoles hiérarchiques :

Le routage hiérarchique [19] est considéré comme étant l'approche la plus favorable en termes d'efficacité énergétique. Il se base sur le concept « nœud standard – nœud maître » où les nœuds standards acheminent leurs messages à leur maître, lequel les achemine ensuite dans le réseau tout entier via d'autres nœuds maîtres jusqu'à la station de base (sink).

Le point fort de ce type de protocoles est l'agrégation et la fusion des données afin de diminuer le nombre de messages transmis au sink, ce qui implique une meilleure économie d'énergie.

LEACH (Low-Energy Adaptive Clustering Hierarchy) [20]: organise les nœuds en clusters. Chaque cluster contient un cluster head qui est responsable de créer et de maintenir l'algorithme TDMA. Les autres nœuds sont appelés les nœuds membres. Les slots TDMA sont assignés aux nœuds membres pour qu'ils puissent échanger des données avec leur cluster head. Ce dernier agrège les données de ses membres et les transmet à la station de base.

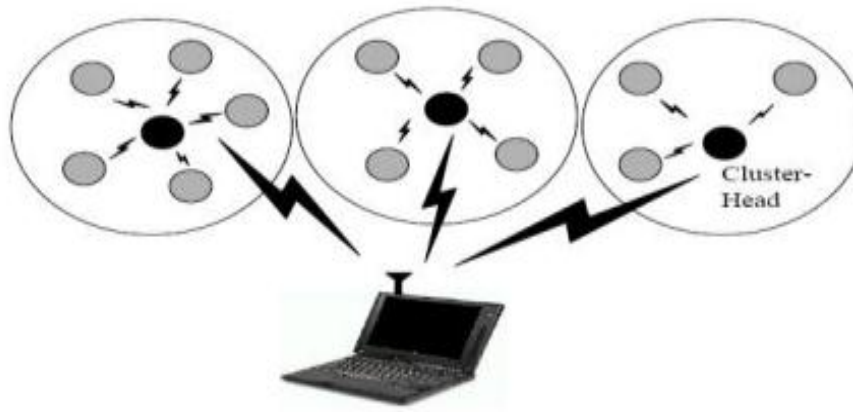


Figure 2.1 : Fonctionnement du protocole LEACH.

Avantage :

- le taux de collision est diminué.
- Lorsqu'un nœud n'est pas dans son slot, il éteint sa radio pour conserver son énergie.

Inconvénient :

- LEACH nécessite une radio complexe qui augmente la consommation d'énergie ainsi que le coût d'un nœud capteur.
- une large consommation d'énergie pour la transmission des messages.

2.4.2. Protocoles dédiés à la couche liaison (sous couche MAC) :

Les sources de consommation d'énergie dans un nœud capteur sont le module radio, le microprocesseur et le capteur [21]. La communication radio est considérée comme la source la plus consommatrice parmi les trois précédentes. Etant donné que la sous couche MAC est principalement concernée par l'utilisation du module radio, plusieurs recherches ont été proposées afin de gérer efficacement cette ressource et optimiser ainsi la consommation énergétique.

En effet, un protocole MAC économe en énergie essaie d'utiliser le module radio le moins souvent possible. L'utilisation inutile du module provient de 5 sources essentielles : l'overhearing, les collisions, l'Idle, les envois infructueux et les messages de contrôle.

Le moyen le plus efficace pour conserver l'énergie est de mettre la radio du nœud capteur en mode veille (low-power) à chaque fois que la communication n'est pas nécessaire. Idéalement, la radio doit être éteinte dès qu'il n'y a plus de données à envoyer et ou à recevoir, et devrait être prête dès qu'un nouveau paquet de données doit être envoyé ou reçu. Ainsi, les nœuds alternent entre périodes actives et sommeil en fonction de l'activité du réseau. Ce comportement est généralement dénommé Duty-cycling ou wake-up scheme. Le mécanisme du Duty-cycling permet de réduire le temps qu'un nœud passe dans l'état d'écoute passive (Idle listening), l'overhearing et d'autres activités inutiles en mettant le nœud dans l'état de sommeil [21].

En effet, de nombreux protocoles MAC introduisent le mécanisme du Duty-cycling dans leurs conceptions afin de réaliser une faible consommation énergétique. Nous classons ces protocoles en deux grandes classes: protocoles synchrones et asynchrones. Les protocoles basés sur le Duty-cycling synchrones sont typiquement équipés avec des périodes d'activation (wake-up schedules) prédéterminées. Chaque période est divisée en une phase de sommeil 'Tsleep' et une phase d'activité 'Tactive'. La synchronisation est effectuée par la transmission fréquente de petites trames de contrôle. Chaque nœud diffuse des trames d'activation une fois qu'il entre dans sa période d'éveil. Ainsi il réveille tous ses nœuds voisins pour une éventuelle communication.

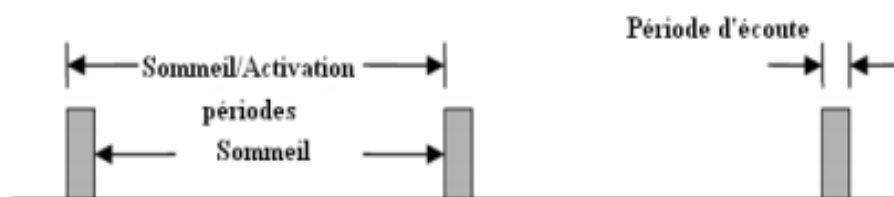


Figure 2.2 : le mécanisme du Duty-cycling.

En ce qui concerne les protocoles basés sur le Duty-cycling asynchrones, on n'est pas obligé de synchroniser les horloges des nœuds capteurs.

Dans ce cas, chaque nœud suit un plan d'activation afin de se mettre en état d'activité ou de sommeil. Ainsi, le plan d'activation arrange les périodes d'activation des nœuds capteurs, de telle sorte à garantir que les nœuds émetteurs et récepteurs s'activent au même instant afin de communiquer leurs données.

S-MAC (Sensor MAC) [22] : est basé sur la méthode CSMA/CA. Il utilise le mécanisme RTS/CTS dans le but de traiter le problème des nœuds cachés. S-MAC introduit une période d'activité et de mise en veille (figure 2.3). Les nœuds doivent être synchronisés pour pouvoir communiquer. Pour ce faire, les nœuds sont organisés en clusters virtuels. Chaque nœud diffuse périodiquement son programme de scheduling dans un paquet SYNC aux autres nœuds dans le même cluster.

Comme dans le cas de IEEE 802.11, la durée de la transmission est connue par le NAV qui est inclus dans les paquets de contrôle RTS et CTS. S-MAC ajoute la durée de la transmission restante dans chaque fragment et dans chaque ACK, cela permettra aux nœuds qui se réveillent au milieu de la transmission de retourner dormir.



Figure 2.3 : La période d'activité et de mise en veille dans S-MAC.

Avantage :

- La perte d'énergie causée par l'écoute au canal libre est réduite par le cycle de mise en veille.

Inconvénients :

- Le protocole n'est pas capable. L'augmentation de nombre de nœuds implique l'augmentation de nombre des schedule sauvegardés dans chaque nœud.
- La latence est augmentée puisque les messages doivent attendre la période d'écoute pour être envoyés.
- L'utilisation d'une période d'écoute fixe cause l'écoute à un canal libre s'il n'y a pas de données à transmettre pendant cette période.

T-MAC (Timeout MAC) : Dans T-MAC, chaque nœud se réveille périodiquement pour communiquer avec ses voisins. Les nœuds communiquent entre eux en utilisant un mécanisme de RTS/CTS qui permet d'éviter les collisions et le problème de station cachée (Voir détaillé dans le chapitre 03).

L'objectif de T-MAC est de pallier aux inconvénients soulevé avec S-MAC en rendant la période d'ordonnancement de frame variable, en fonction du niveau de trafic. Chaque nœud se réveille périodiquement, écoute le canal et reste éveillé durant une période, variable et déterminé dynamiquement, avant de retourner ensuite en mode sommeil (voir Figure 2.4).

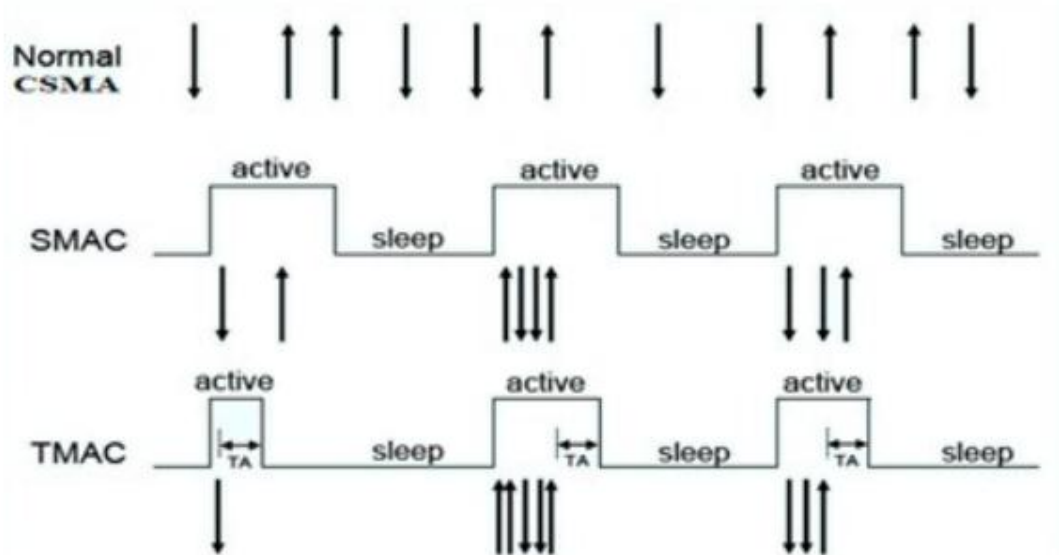


Figure 2.4 : Sommeil et réveil cycles SMAC et TMAC.

B-MAC (Berkeley MAC) : est basé principalement sur deux principes : l'analyse du bruit sur le canal radio et sur l'écoute basse consommation [23]. Quand un nœud désire envoyer un paquet, il détermine si le canal radio est utilisé par un autre nœud ou pas en écoutant le "bruit" en se basant sur un indicateur de puissance du signal. S'il n'y a pas de bruit, le canal est libre et il peut donc émettre. Avant d'envoyer des données, il doit émettre un préambule (voir Figure 2.5).

Les nœuds sont en mode sommeil la plupart du temps et se réveillent à intervalles réguliers. A leur réveil, ils écoutent le bruit sur le canal radio. S'il n'y a pas de bruit sur celui-ci, le nœud retourne en mode sommeil. S'il y a du bruit, cela signifie que des données vont arriver (à cause du préambule). Le préambule doit être suffisamment long (au moins égal à la période de sommeil), pour que tous les nœuds (et donc les destinataires des données) puissent l'entendre. Ensuite, quand les données commencent à arriver, les nœuds qui ne sont pas destinataires de ces données retournent en mode sommeil. B-MAC ne nécessite pas une synchronisation pour que les nœuds dorment et se réveillent en même temps.

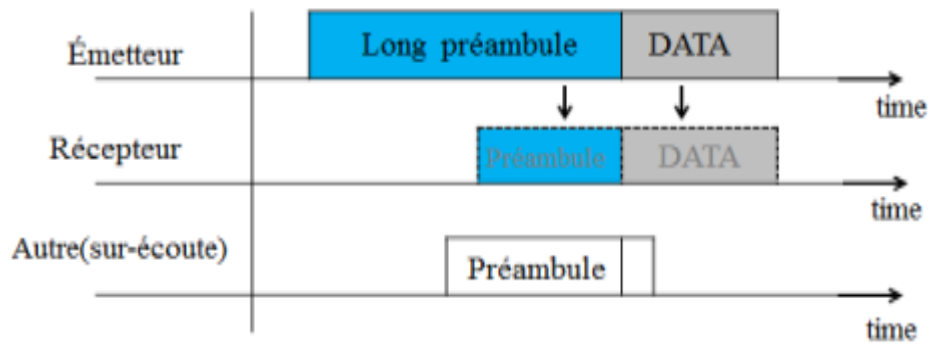


Figure 2.5 : Protocole B-MAC.

2.5. CONCLUSION

Dans ce chapitre, Nous avons essayé à travers le point sur les différents protocoles de routage dédiés aux RCSFs, nous avons vu quelques protocoles MAC les plus étudiés dans la littérature, pour les réseaux de capteurs sans fil.

L'économie d'énergie constitue un défi très important pour la conception d'RCSFs robustes et sécurisés contre les défaillances de ressources. Vu que la disponibilité de ces derniers est directement liée à la durée de vie de leurs batteries, l'application des protocoles de communication à faible consommation d'énergie est plus que nécessaire. Les protocoles mono couche ont prouvé leur efficacité en termes d'économie d'énergie au niveau de leurs couches respectives.

3.1. Introduction :

Les réseaux de capteurs sans fil ont un aspect supplémentaire : en tant que nœuds de capteurs sont généralement fonctionnant sur batterie, la consommation d'énergie est très importante. La radio sur un nœud de capteur est généralement le composant qui utilise plus d'énergie. Non seulement la transmission des coûts de l'énergie ; de réception, ou simplement l'éther balayage pour la communication, peut utiliser jusqu'à la moitié, en fonction du type de radio.

Plusieurs protocoles de couche mac ont été proposées dans la littérature et mis en œuvre dans des applications réelles dans les réseaux de capteurs sans fil. T-MAC est argument fondé mac protocole réseau simple et à haute efficacité énergétique mise en œuvre. Vie du réseau, la consommation d'énergie, latence et une transmission fiable des paquets sont quelques-uns des principaux problèmes dans les réseaux de capteurs sans fil. Depuis la batterie, la mémoire et la capacité de traitement sont limitées ; Il y a nécessité de très efficace Protocole de transmission de couche Mac. Dans cet article, nous vous proposons un nouveau protocole mac qui utilise les caractéristiques intrinsèques du protocole T-MAC et il optimise avec une fonction de réduction simple. Cette disposition simple découpe la période extra Ecoute qui a une probabilité plus élevée d'obtenir des déchets pendant la durée d'Activation [29].

T-MAC (Time-Out-MAC) est un protocole qui est dérivé du protocole S-MAC, dans lequel le non sommeil et de veille périodes sont fixes [30]. T-MAC dans le nœud de capteur dévie pour dormir période si aucun événement n'a eu lieu pendant un temps «Tact» comme représenté sur la Figure 3.1.

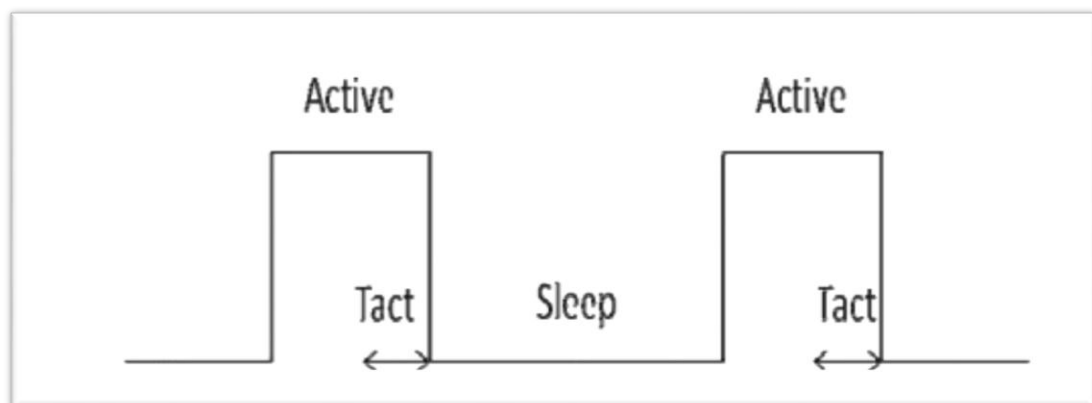


Figure 3.1 : T-MAC Protocol.

Il existe de nombreux événements tels que la réception de données, début de l'écoute ou de la période de sommeil etc. ralenti minimum de temps d'écoute est le temps «Tact». L'intervalle T_a est supérieure à la somme des temps de contention, la longueur d'un paquet RTS, retournement le temps et la longueur du paquet CTS.

Ce scénario se traduit par une consommation d'énergie inférieure à T-MAC par rapport au protocole de capteur S-MAC. Toutefois, cela est réglé contre haute retard ou une latence qui protocole T-MAC a par rapport au protocole S-MAC [30].

Dans T-MAC, chaque nœud se réveille périodiquement pour communiquer avec ses voisins [31]. On a donc, comme dans S-MAC des périodes actives et inactives. Les nœuds communiquent entre eux en utilisant un mécanisme de RTS/CTS (Request-To-Send, Clear-ToSend) qui permet d'éviter les collisions et le problème de station cachée. En T-MAC tous les messages sont transmis dans un éclat de longueur variable et il y a écart entre les salves appelés sommeil / temps de sommeil.

Cela permet de réduire l'écoute au ralenti. Le nœud se réveille périodiquement pour communiquer avec les voisins et il utilise RTS et CTS, données accusé de réception (ACK) régime, qui fournit à la fois la prévention des collisions et fiable transmission. Un nœud écoute le canal radio et peut potentiellement transmettre tant qu'il est dans la période active. Une période active se termine quand aucun événement d'activation ne se produit pendant un temps T_A .

3.2. T-MAC PROTOCOL DESIGN :

Consommation d'énergie est le Critérium de conception principale pour notre conception des protocoles MAC [32]. Nous ont déjà identifié le problème de ralenti à l'écoute. Autres formes d'énergie déchets sont :

Collisions : si deux nœuds de transmettent en même temps et interfèrent avec chacun d'autres transmission, les paquets sont corrompus. Par conséquent, l'énergie utilisée au cours de transmission et la réception est perdu.

Protocol overhead : généraux la plupart des protocoles nécessitent des paquets de contrôle devant être échangé ; qu'ils ne contiennent aucune données d'application, nous considérons toute l'énergie utilisée pour transmettre et recevoir ces paquets comme frais généraux.

overhearing: car l'air est un milieu partagé, un nœud peut recevoir des paquets qui ne sont pas destinés il pourrait alors aussi bien ont éteint sa radio.

Le problème est que, si les exigences de latence et espace de mémoire tampon sont généralement fixe, le taux de messages oscilleront. Si les messages importants ne doivent ne pas être manqués et sans importance Messages aurait pas dû être envoyé dans tous les cas, les nœuds doivent être déployés avec une temps d'activité qui peut gérer la charge prévue la plus élevée. Chaque fois que la charge est plus faible que celle, le temps actif ne sert pas optimale et l'énergie sera perdu sur le ralenti à l'écoute. L'idée originale du protocole T-MAC est de réduire l'écoute ralenti par transmettre tous les messages en rafales de longueur variable et dormir entre éclate.

Pour maintenir un temps actif optimal sous charge variable, nous dynamiquement déterminer sa longueur [32]. Nous terminons le temps actif de manière intuitive : Time out sur rien entendre.

3.3. Basic Protocol :

La figure 3.2 illustre le régime de base du protocole T-MAC. Chaque nœud se réveille périodiquement pour communiquer avec ses voisins et se dirige ensuite vers dormir à nouveau jusqu'à la section suivante. Pendant ce temps, les nouveaux messages sont en attente. Les nœuds communiquent entre eux à l'aide d'une demande-à-Send (RTS), Système de Clear-To-Send (CTS), données, accusé de réception (ACK), qui prévoit les collisions et une transmission fiable [33].

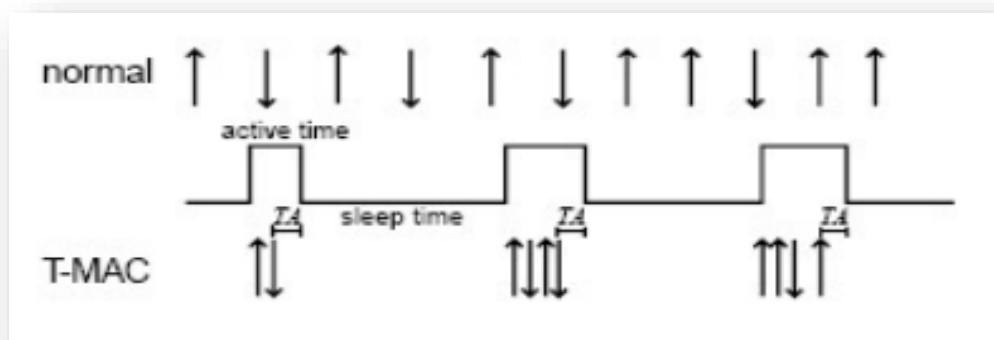


Figure 3.2 : Le schéma de protocole base T-MAC, avec adaptation actives moments.

Un nœud gardera à l'écoute et potentiellement transmettre, tant que c'est dans une période active. Une période active se termine lorsque aucun événement d'activation s'est produit pour une fois TA. Un événement d'activation est :

- la mise à feu d'une minuterie de trame périodique ;
- la réception de toute donnée sur la radio ;
- la détection de communication¹ sur la radio, par exemple lors d'une collision ;
- la fin d'émission d'un paquet de données propres du nœud ou d'accusé de réception ;
- les connaissances, acquis par le biais de surprendre les paquets RTS et CTS préalables, qu'un échange de données d'un voisin est terminé.

Un nœud va dormir s'il est inactif et non dans une période active. Notez que TA est une limite supérieure sur la durée d'inactivité écoute par image à la fin du temps actif.

Le schéma décrit timeout déplace toutes les communications d'un éclat au début de la trame. Puisque les messages entre temps actifs doivent être mise en mémoire tampon, la capacité de la mémoire tampon détermine une limite supérieure sur le maximum temps de frame. Voici les trois principales caractéristiques de T-MAC [34] protocole :

- ❖ Il réduit idle listening en transmettant tous les messages dans une rafale de longueur variable et de dormir entre les salves.
- ❖ Afin de maintenir un temps actif optimal sous une charge variable à la longueur des données à transmettre est déterminé dynamiquement.
- ❖ Le temps actif d'un nœud se termine lorsqu'aucun événement n'est entendu sur une période de temps déterminée.

3.4. Clustering and synchronisation :

Synchronisation de frame s'inspire des clusters virtuels, tels que décrits par les auteurs du protocole S-MAC [32]. Lorsqu'un nœud s'anime, il commence par en attente et écoute. Si elle n'entend rien pour un certain laps de temps, il choisit une planification frame et transmet un paquet SYNC, qui contient le temps jusqu'à ce que le frame suivant commence. Si le nœud, au démarrage, entend une SYNC paquet d'un autre nœud, il suit la planification dans ce paquet SYNC et transmet ses propres SYNC en conséquence.

Nœuds de retransmettent leur SYNC de temps en temps. Nœuds doivent également écouté un remplir le frame sporadiquement, donc ils peuvent détecter l'existence des différentes planifications. Cela permet aux nœuds mobiles et nouveaux pour s'adapter à un groupe existant.

Si un nœud possède une planification et entend une SYNC avec une planification différente de un autre nœud, il doit adopter les deux planifications. Il doit également transmettre une SYNC avec son propre planification sur l'autre nœud, pour informer l'autre nœud sur la présence d'une autre planification. Adoptant les deux planifications signifie que le nœud a un événement d'activation au début de ces deux cadres.

Nœuds doivent commencer une transmission de données seulement au début de leur propre temps d'activité. A cette époque, les voisins avec la même planification et les voisins qui ont adopté la planification comme extra, sont éveillés. Si un nœud commencerait à transmission au début du frame d'un voisin, il pourrait émettre à l'autre, voisin de sommeil. Notez que ce schéma rend possible cette émissions suffit à transmettre une fois.

Le schéma décrit la synchronisation, qui est appelé Virtual clusters [32], demande instamment aux nœuds pour former des grappes selon l'horaire, sans respecter cela l'annexe de tous les nœuds du réseau. Il permet la diffusion efficace, et élimine la nécessité de maintenir des informations sur différents voisins.

La technique de gestion de clusters virtuelle est facile à implémenter. Conserver plusieurs horaires avec un temps actif de longueur fixe est plus complexe, depuis les temps actifs se chevauchent.

3.5. Détermination de TA :

Un nœud ne doit pas s'endormir (entrer en phase « sleep » du Schedule) pendant que ces voisins communiquent puisqu'il peut être à tout moment le récepteur d'un message. La réception d'un RTS ou d'un CTS est nécessaire pour actualiser l'intervalle TA [35].

Du fait qu'un nœud C peut ne pas entendre, en raison de la portée limitée d'émission, un paquet RTS émis par un nœud A, l'intervalle TA doit être suffisamment large pour que C puisse recevoir au minimum la date de l'émission du paquet CTS

(entre la réception d'un paquet et la date de son émission, il ya une différence du fait que le récepteur est à l'écoute du support).

La longueur de l'intervalle TA doit vérifier l'inégalité suivante :

$$TA > C + R + T$$

Avec :

C : longueur de l'intervalle de contention.

R : durée de transmission du paquet RTS.

T : intervalle de temps (très petit) entre la fin de réception du paquet RTS par B et le début d'émission du paquet CTS par B.

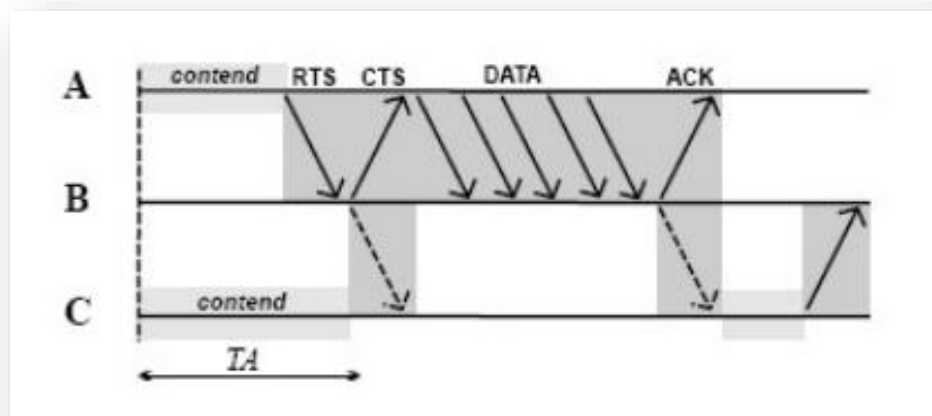


Figure 3.4 : Choix du paramètre TA dans le protocole Timeout-MAC.

Le fait de choisir un intervalle TA très large peut avoir des conséquences négatives puisque l'augmentation du TA implique une augmentation de la phase d'idle listening donc augmentation de la consommation d'énergie.

La valeur de TA qui a été adoptée dans l'implémentation de TMAC dans les nœuds EYES est $TA = 1.5 \cdot (C + R + T)$.

3.6. Overhearing Avoidance :

Chaque nœud maintient un vecteur d'allocation du réseau (network allocation vector NAV) dans lequel il indique l'activité de ses voisins. Chaque paquet de données contient un champ indiquant le temps restant [36].

Lorsqu'un nœud reçoit un paquet de données qui ne lui est pas destiné, il met à jour son NAV. A chaque intervalle de temps, la valeur du NAV est décrétementée (si elle est non nulle). Ainsi une valeur non nulle du NAV signifie qu'il y a une transmission. Le nœud peut donc s'endormir et ne se réveiller ensuite que lorsque le NAV devient nul.

3.7. L'évitement du phénomène d'Overhearing :

Cette idée a été introduite par le protocole SMAC et qui consiste pour un nœud A à entrer en phase « sleeping » [36] dès qu'il écoute un RTS ou CTS destiné à un autre nœud. Durant cette phase, le nœud A peut mettre son radio en « turn-off » pour sauvegarder de l'énergie. Cependant, le protocole T-MAC quant à lui, considère l'évitement d'Overhearing comme une option du fait qu'il peut perturber les communications entre les nœuds dans le cas d'application qui requiert le débit maximum.

En effet, pendant la période où le nœud A va entrer en phase « sleeping », il peut manquer la réception de paquets RTS ou CTS destinés vers lui et va donc perturber les communications avec autrui lors de son réveil. Par conséquent, le débit va diminuer. Malgré que l'évitement de l'Overhearing soit un phénomène qui économise de l'énergie (cas de SMAC), il n'est pas conseillé de l'utiliser dans les applications où on veut assurer un débit maximum.

3.8. Le problème du « early-sleeping » :

On « supposera » que le trafic dans le réseau est du type unidirectionnel, entre autre les communications entre les nœuds et le Sink [36]. Tout ce qui émane de A est vu par B, tout ce qui émane par B est vu par A et C, par C est vu par B et D... Ce problème est illustré par la figure suivante :

D va dormir avant que C puisse lui envoyer un RTS. On considère le nœud C qui, à chaque fois où il veut transmettre des données vers D, doit conventionner le médium. Cette contention peut être perdue par C dans deux cas :

- S'il reçoit un paquet RTS de B.
- S'il « sur-entend » un paquet CTS émis par B vers A.

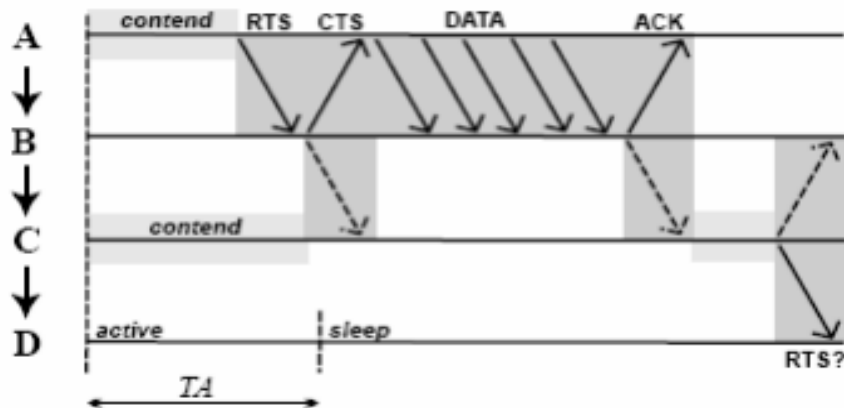


Figure 3.5 : Le problème du « early-sleeping ».

Dans le premier cas, C va répondre B par un paquet CTS qui va être « sur-entendu » par D. Ainsi, D va reconnaître que son voisin C est entré en communication avec B, ce qui entraînera D à s'endormir jusqu'à ce que B et C finissent leur communication. Dans le deuxième cas, C doit rester silencieux, jusqu'à ce que A et B finissent leur communication, et ceci pour ne pas perturber leur communication.

Le fait que D n'entend aucune réaction de la part de C, va engendrer D à terminer son intervalle TA et à s'endormir. Ce n'est qu'au prochain état actif (prochain frame) de D que le nœud C aura une chance d'émettre son paquet RTS vers D.

3.9. Communication asymétrique :

Des expériences préliminaires ont révélé un problème avec le protocole de T-MAC lorsque le trafic à travers le réseau est le plus souvent unidirectionnel, comme dans un évier de nœuds modèle de communication [35].

Chacun des nœuds A si D dans la forme de photo une cellule avec ses voisins. Les messages circulent de haut en bas, alors nœud A envoie à b, B qu'à C et C seulement à D. Considérons maintenant le nœud C. Chaque fois qu'il veut envoyer un message à D, il doit faire face à moyen et risquez de perdre des nœuds B (par la réception d'un paquet RTS) ou nœud A (indirectement, par surprennent un paquet CTS de nœud B). Si le nœud C perd la contention en raison d'un paquet RTS de nœud B, il sera réponse avec un paquet de CTS, qui peut également être entendu par nœud D. Dans ce cas, le nœud D sera éveillé lorsque la communication entre C et B se termine.

Toutefois, si le nœud C perd l'affirmation parce qu'il surprend un paquet de CTS de B à A, C doit garder le silence. Puisque D ne connaît pas la communication entre A et B, son temps d'activité prendra fin, et nœud D mettra pour dormir. Seulement au début de la prochaine trame nœud C aura une nouvelle chance à Envoyer au nœud D.

Nous appelons l'effet observé le problème de sommeil au début (car un nœud va à dormir quand un voisin a toujours des messages pour lui). Dans les nœuds-à-évier modèle de communication, le problème de sommeil au début réduit le débit total possible de T-MAC à moins de la moitié du débit maximal de les protocoles traditionnels ou S-Mac.

Dans les expériences plus tard [37], nous avons également rencontré ce problème à la frontière d'une partie très active du réseau. Nous pensons que le problème peut se produire dans toute communication asymétrique modèle.

Nous avons proposé deux solutions à ce problème qui sont :

- FRTS (Future Request-to-send).
- Taking priority on full buffers.

3.9.1. FRTS (Future Request-to-send):

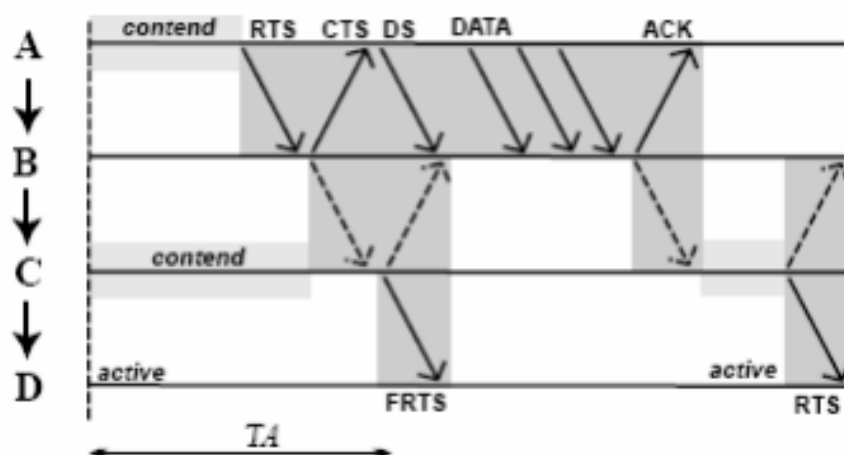


Figure 3.6 : L'échange de paquet Future Request To Send (FRTS).

L'échange de paquet Future Request To Send (FRTS) va garder le nœud D réveillé. Si un nœud C “sur-entend” un CTS (2ème cas énoncé précédemment), il pourra immédiatement émettre un paquet FRTS contenant la durée de l'échange entre A et B (cette durée est connue grâce au paquet CTS) vers D [37].

Le nœud récepteur de ce paquet FRTS (D) va reconnaître qu'il sera le prochain récepteur d'un paquet RTS, il devra alors rester à l'état actif pour ne pas manquer ce paquet RTS que C souhaite lui émettre. Afin que ce paquet FRTS n'entre pas en collision avec les paquets DATA qui seront émis par A suite à sa réception du paquet CTS provenant de C, les paquets DATA doivent être envoyés juste après que le paquet FRTS sera reçu par D.

Afin de prévenir n'importe quel autre nœud de prendre le canal pendant cette période, A va transmettre un paquet DS (Data-Send) [37] à la suite duquel il va émettre ses paquets DATA. Le paquet DS va entrer en collision avec le paquet FRTS « sur-écouté » par B ce qui entraînera la destruction de DS. Ce résultat n'est pas grave du fait que le paquet DS ne contient pas d'informations utiles.

3.9.2. Taking priority on full buffers:

Lorsqu'un nœud possède un buffer plein, il est plus préférable pour lui d'émettre que de recevoir. Dans le cas où C recevra un RTS, il devra émettre son propre RTS vers D au lieu de répondre B par un CTS.

Ceci peut avoir deux conséquences; [37] La première est que C aura une chance assez élevée pour émettre un paquet RTS vers D avant que ce dernier ne va s'endormir (s'il reçoit un CTS de la part de C : cas énoncé précédemment alors D va s'endormir) La deuxième conséquence est un contrôle de flux assez avantageux dans le cas de communications entre les nœuds et le Sink. En effet, le nœud B va être prévenu de l'émission de paquets vers C jusqu'à ce que ce dernier ait suffisamment de place dans le buffer.

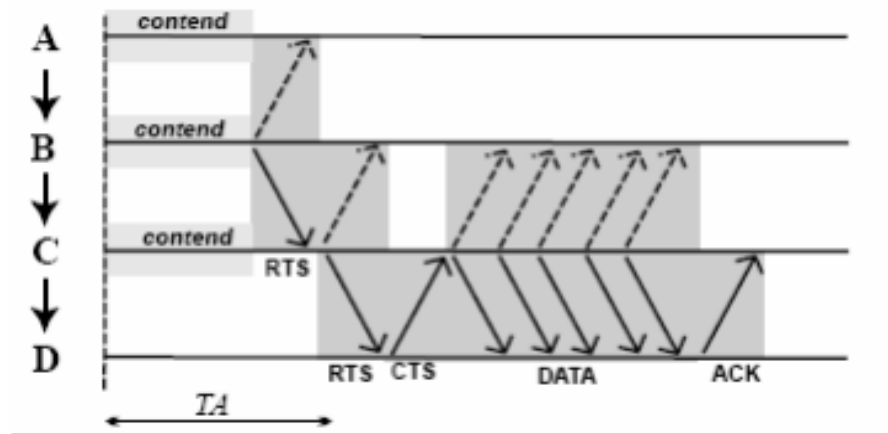


Figure 3.7 : Prise de la priorité.

Le protocole T-MAC emploie une auto-configuration pour identifier les nœuds voisins lorsque le réseau est initialisé et permettre l'ajout et la suppression de nœuds. T-MAC ajuste sa consommation d'énergie [37], dans le contexte d'un trafic variable, grâce au Time-Out Timer (TA).

Les avantages et Les inconvénients de T-MAC :

Les réductions de T-MAC dans la consommation d'énergie sont très prometteuses. Ce nouveau protocole fait l'objet d'une étude en cours, et nous nous attendons à des résultats plus détaillés dans l'avenir.

Avantage:

- ✓ Délai est réduit au minimum. Il dépasse également les autres protocoles MAC sous charge variable.
- ✓ T-MAC peut facilement manipuler des charges variables en raison de l'horaire de sommeil dynamique.
- ✓ T-MAC conserve l'énergie puisqu'il évite l'écoute à un canal libre même dans la période d'écoute.
- ✓ T-MAC est simple.
- ✓ T-MAC contient beaucoup de messages de contrôle (overhead).

Inconvénients:

- ✓ Problème de sommeil tôt dans lequel les nœuds peuvent dormir selon leur temps d'activation.
- ✓ Les données peuvent être perdues en particulier pour les longs messages.
- ✓ T-MAC n'est pas capable.
- ✓ La diffusion de paquets n'utilise pas le mécanisme RTS/CTS, ce qui augmente.
- ✓ la probabilité de collisions.

3.10. Conclusion

Conception d'un protocole MAC qui permet d'améliorer l'efficacité énergétique d'étendre vie du réseau en WSN est un problème difficile en raison de ressources strictes limitation en nœuds de capteurs et de la particularité des médias sans fil.

Nous avons présenté un protocole MAC pour réseaux de capteurs sans fil, T-MAC qui économise l'énergie en éteignant la radio autant que possible. Le moment d'éteindre la radio est déterminé par un délai d'attente.

Pour résoudre le problème d'écoute inactif dans un réseau de capteurs sans fil, T-MAC adapte dynamiquement un devoir d'Ecoute/sommeil cycle d'une nouvelle façon, grâce à des délais d'attente à grains fins, tout en ayant moins complexité.

4.1. Introduction :

Les technologies informatiques s'étendent de plus en plus ce qui mène à l'apparition de plusieurs systèmes et ainsi différentes méthodes de gestion. Dans le modèle réseau, l'étude passe essentiellement par l'étude de ses composants atomiques. Ce principe est forcément appliqué pour les réseaux de capteurs sans fil, les différentes propositions des chercheurs pour remédier aux problèmes des RCSFs tel que la consommation d'énergie, la durée de fonctionnement du réseau, les contraintes de collecte d'informations... nécessite une phase de test avant la mise en place. Pour cela, la solution la plus fiable et la moins coûteuse consiste en " La Simulation " [38].

La simulation des réseaux de capteurs consiste principalement en la reproduction du comportement et du fonctionnement des nœuds capteurs dans un environnement informatique ; pour des raisons tel que : La répétition d'expérience, L'adressage des systèmes complexes, Le gain de temps et la variation des paramètres de simulation alors que la simulation réelle s'avère coûteuse, voir impossible dans quelque cas.

Dans ce qui suit, nous parlerons brièvement des simulateurs de réseaux existants, du simulateur choisi pour nos simulations "OMNET++" et de la plate forme des simulations de réseaux de capteur sans fil "CASTALIA".

4.2. Généralités sur la simulation :

Dans nos jours, la simulation connaît un essor considérable, et ce grâce à l'intérêt que présente les modèles informatiques des systèmes simulés ;La simulation consiste à la modélisation informatique d'un système quelconque, en offrant une représentation de toutes les entités de ce système, leurs comportements propres, ainsi que leurs interactions. Elle met à la disposition de l'utilisateur un environnement d'expérimentation dont on peut faire varier les paramètres [38].

Grâce aux progrès réalisés dans le domaine du développement et des techniques de programmation, nous disposons aujourd'hui de langages de programmation très puissants. Ainsi, il devient possible de réaliser un simulateur dans un environnement de programmation existant.

4.3. Types de simulation :

En fonction du type d'évènements dans la simulation [38], nous distinguons deux types de systèmes de simulation : les systèmes discrets et les systèmes continus.

- **Systèmes de simulation discrète :** Sont des systèmes pour lesquels les variables concernées par la simulation ne changent d'état qu'en un nombre fini de points sur l'axe du temps. On appelle également ces systèmes : systèmes de simulation à évènements discrets.
- **Systèmes de simulation continue :** Ce sont des systèmes pour lesquels les variables peuvent changer d'état à n'importe quel instant pendant la simulation.

4.4. Les simulator de réseau existant :

Il existe plusieurs simulateurs de réseau tel que : NS2, OMNET++, OPNET, GLOMOSIM, JSIM...etc. [39] Parmi ceux, on va citer quelques simulateurs comme NS2, GLOMO-SIM et bien sur notre simulateur OMNET++.

4.4.1. NS2 :

NS2 est un simulateur à évènements discrets très répandu dans le domaine de la recherche pour tout ce qui se réfère aux réseaux. L'utilisation de l'appellation "NS2" précise la version du simulateur NS. Il constitue un support important pour la simulation de protocoles TCP, protocoles de routage, protocoles de multicast.

Le développement de NS suit une approche orientée objet qui utilise deux langages de programmation : C++ et TCL. Les modules de base du simulateur et les protocoles sont implémentés en C++ avec une couche TCL au-dessus, qui fournit une interface flexible et facile à utiliser [40]. La figure suivante illustre les différents composants internes du simulateur :

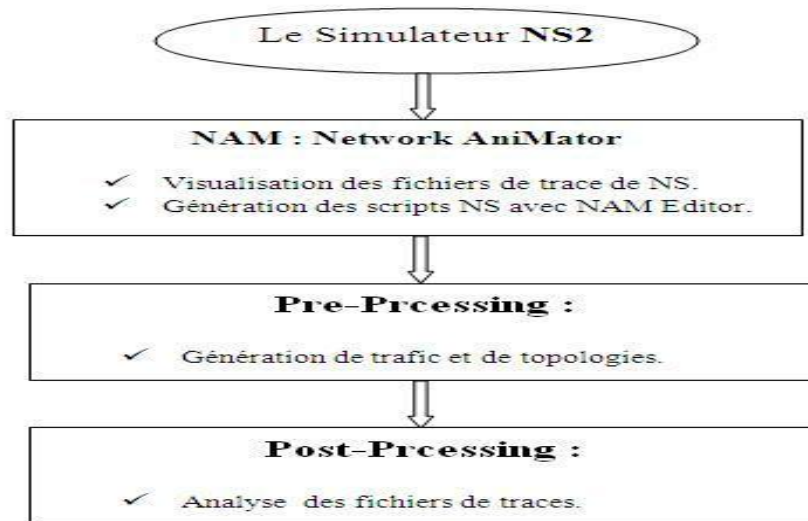


Figure 4.1 : Description architecturale du simulateur NS2.

4.4.2. GloMoSim

GloMoSim (Global Mobile information system Simulator) a été conçu selon une architecture orientée « couche » similaire que celle des sept couches OSI pour les réseaux [41]. Il a été développé au laboratoire UCLA Parallel Computing Laboratory en utilisant le langage PARSEC.

PARSEC (PARallel Simulation Environnement for Complex system) est un langage de programmation dérivé du langage MAISIE. Tous deux sont destinés à la simulation à événements discrets. Ils ont été développés à partir du langage C à l'université UCLA (University California Los Angeles). Le transfert de paquets dans GloMoSim se fait selon le schéma suivant :

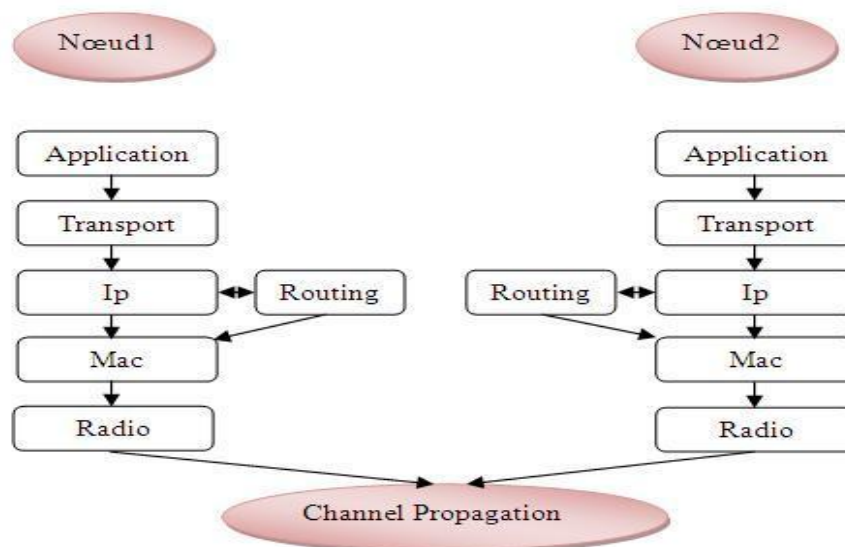


Figure 4.2 : Transfert des paquets dans GloMoSim.

4.4.3. OMNeT++

OMNeT++ IDE (Integrated Development Environment) est basé sur la plateforme Eclipse. C'est un environnement open source qui fournit des outils pour la création et la configuration des modèles de réseaux (les fichiers NED et INI) et des outils pour l'exécution d'un lot de programmes ainsi que pour l'analyse des résultats de simulation. OMNeT++ semble séduire de plus en plus la communauté scientifique et un nombre croissant de modèles sont disponibles [42]. Dans ce qui suit, il aura une vue plus détaillée sur le simulateur ainsi les raisons du choix de ce simulateur.



Figure 4.3 : Le lancement du simulateur Omnet++.

4.5. Comparaison entre les simulateurs

La simulation est souvent moins chère que l'expérimentation et comporte beaucoup moins de risques lorsque l'homme fait partie du système étudié. Les résultats peuvent être obtenus beaucoup plus rapidement. La simulation (surtout numérique) est basée sur une connaissance des phénomènes qui ne peut être obtenue que par l'expérimentation. Une simulation ne peut donc être réalisée que si on dispose d'un acquis de connaissances suffisant obtenues par des expérimentations sur des phénomènes antérieurs et analogues. Quelle que soit la qualité de la simulation, elle ne remplace pas totalement l'expérimentation [43].

Simulateur	Avantages	inconvénients
NS2	<ul style="list-style-type: none"> • Utilise l'Orienté Objet • Un grand nombre de protocoles disponibles publiquement. • La disponibilité d'un outil de visualisation. • Simulateur multi couches 	<ul style="list-style-type: none"> • Conçu pour les réseaux filaires. • Support seulement deux protocoles MAC sans fil, 802.11, et TDMA. • Difficulté d'ajout de nouveaux modèles à cause des dépendances entre modules. • Scénario de simulation décrit en OTCL.
GloMoSim	<ul style="list-style-type: none"> • La disponibilité d'un outil de visualisation. • Initialement conçu pour les réseaux sans fil (surtout Ad Hoc). • Exécution des simulations relativement rapide. 	<ul style="list-style-type: none"> • Nécessité de maîtriser le Parsec pour toute personnalisation autour du noyau. • Installation assez complexe.
Omnet++	<ul style="list-style-type: none"> • Architecture modulaire permettant l'intégration de nouveaux modèles. • Utilisation de C++ (et récemment C#) pour le développement du noyau. • Les classes de base du simulateur peuvent être étendus et personnalisés. • Conception de modèles se rapprochant de la réalité. 	<ul style="list-style-type: none"> • Ne prend pas en charge le cas des réseaux de capteurs. • Peu de modèles pour les sans fils. • Description des modèles en langage NED.

Tableau 4.1 : les avantages et inconvénients des simulateurs décrit auparavant.

4.6. Le Simulator OMNET++ (Objective Modular Network Testbed in C++):

4.6.1. Choix du simulateur OMNET++

On a choisi le simulateur OMNET++, a cause de son architecture modulaire qui permettre l'intégration de nouveaux modèles, et pour notre sujet on va implémenter un nouveaux modèle spécifique aux réseaux de capteurs.

Le déploiement d'un réseau de capteurs exige une étape de simulation avant son installation sur site. La simulation permet de tester à moindre coût les performances d'une solution. OMNeT++ est un environnement de simulation à événements discrets basé sur le langage C++, une application open source et sous licence GNU [44]. Il est totalement programmable, paramétrable et modulaire ainsi grâce à son architecture flexible et générique, il a été utilisé avec succès dans divers domaines, notamment :

- La modélisation de réseaux de file d'attente,
- La modélisation de protocoles de communication,
- La validation des architectures hardware,
- L'évaluation de performances pour des systèmes software complexes.

OMNET++ sera notre environnement de simulation, grâce à son architecture modulaire, il s'agira d'étendre le simulateur en implémentant un nouveau modèle spécifique aux réseaux de capteurs.

4.6.2. Description architecturale d'OMNET++

Les modèles OMNET++ constituent en un ensemble de modules hiérarchiquement emboîtés tel qu'il est montré dans la (Figure III.4) :

Les modules Simples sont écrits en C++ en utilisant la librairie de simulation d'OMNET++, Ces derniers contiennent des algorithmes relatifs au modèle implémenté. Le groupement des modules simples constitue des modules composés sachant que leurs communications sont gérées grâce à des connexions entre les modules via des " gâtes (ports) ".

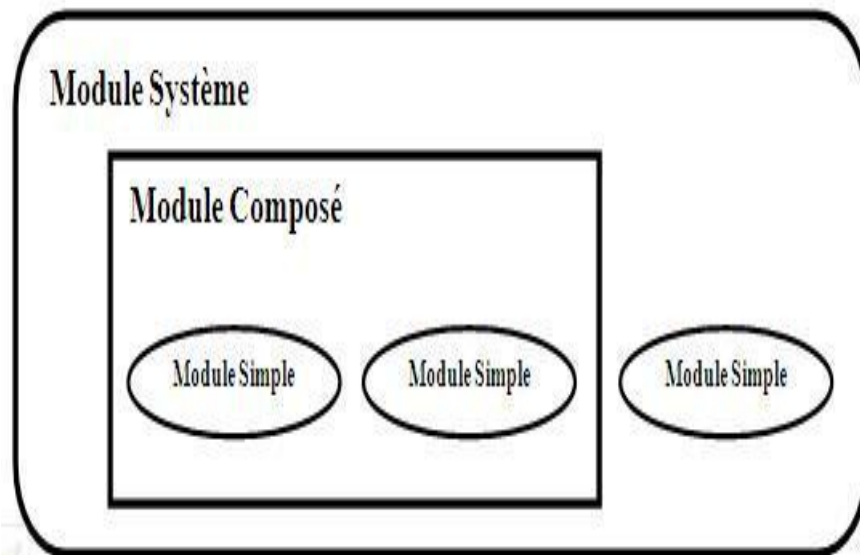


Figure 4.4 : Architecture modulaire du simulateur Omnet++.

Au niveau plus élevé, le module système est créé par l'utilisateur. C'est un module spécial qui n'a pas de connexions avec l'environnement extérieur, mais plutôt avec ses composants internes (Modules simples et composés) [45]. Les modules peuvent s'attribuer des paramètres assignés aux modules dans les fichiers de description de réseaux (fichiers NED) ou encore dans le fichier de configuration « omnetpp.ini ». Ces paramètres sont utiles pour la personnalisation du comportement des modules simples ou encore pour le paramétrage de la topologie du modèle.

6.4 Installation du simulateur OMNET++

L'installation d'OMNET++ se fait en différentes étapes suivant une procédure d'installation décrite dans le package téléchargé [46] selon le système d'exploitation installé. Les éléments installés sur l'ordinateur seront les suivants :

- ✚ Une bibliothèque de simulation interne.
- ✚ Un compilateur du langage descriptif de la topologie NED (nec).
- ✚ Un éditeur de réseaux graphiques pour les fichiers NED (GNED).
- ✚ Un exécutable Omet++.
- ✚ Une Interface graphique de simulation IDE.
- ✚ Un outil de documentation de modèle (opp_neddoc).
- ✚ Autres utilitaires (l'outil de création makefile, etc.).
- ✚ Une documentation, des simulations types, etc.

4.6.3. Les principaux fichiers d'OMNET++

Les différents fichiers sont [44] :

- **Fichier (.Ned)** : Utilise le langage NED de description de réseau. Il peut être utilisé en 2 modes : Mode Graphique ou Mode Texte qui permettent de décrire les paramètres et les ports du module. Les erreurs commises sont indiquées en temps réel par un point rouge situé à la gauche du code.
- **Fichier (.ini)** : Est lié étroitement avec le fichier NED. Permet à l'utilisateur d'initialiser les paramètres des différents modules ainsi la topologie du réseau.
- **Fichier (.msg)** : Les modules communiquent en échangeant des messages. Ces derniers peuvent être déclarés dans un fichier dont l'extension est (.ms) où l'on peut ajouter des champs de données. Omet++ traduira les définitions de messages en classes C++. Le diagramme suivant peut donner une idée plus détaillée sur le développement d'exécution d'une simulation sous Omnet.

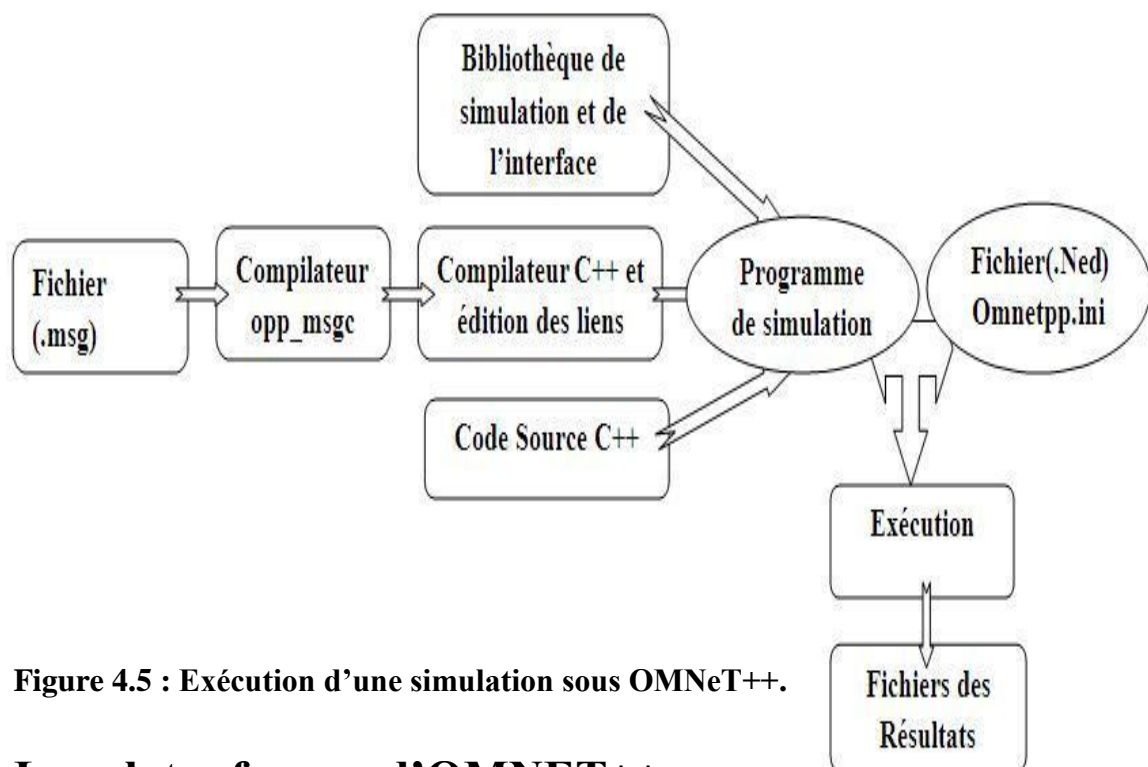


Figure 4.5 : Exécution d'une simulation sous OMNeT++.

4.7. Les plates formes d'OMNET++

Le simulateur OMNeT++ n'est pas spécialisé pour les réseaux de capteurs sans fils, Pour cela il existe plusieurs extensions, plate forme et simulateurs basés sur OMNET++ qui essayent d'introduire ce manque comme "Mixim", "Castalia".

4.7.1. Mixim : est un simulateur [47] qui intègre et développe plusieurs cadres existants pour les simulations sans fil et mobiles dans OMNeT++. Bien que OMNeT++ n'a pas un soutien clair des modèles de simulation pour la communication sans fil. Mixim fournit des modèles détaillés de canal sans fil (décoloration, etc.), la connectivité sans fil, la mobilité, les obstacles et les protocoles MAC comme IEEE 802.11b et IEEE 802.15.4. Mixim propose des modules pour une mise en œuvre facile de nouveaux protocoles MAC comme Time Division Multiple Access (TDMA) ou en fonction des protocoles hybrides (p. ex Z-MAC). Mixim est écrit avec le langage C++ comme Castalia et OMNeT++.

4.7.2. Castalia : est également un simulateur [47] de niveau pour les réseaux de capteurs basé sur OMNeT++. Il s'agit d'un simulateur générique avec un canal sans fil et un modèle de radio basé sur des données mesurées. Comme il est illustré à la (Figure III.6), définit trois modules principaux : le nœud, Processus physiques et les canaux sans fil. Castalia est développé avec le C++.

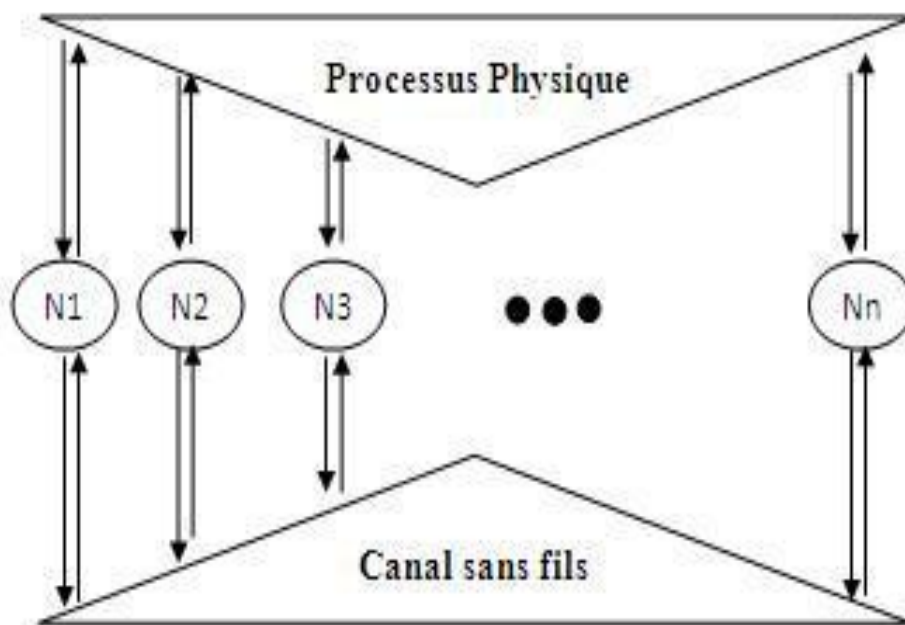


Figure 4.6 : Les connexions des modules sous Castalia.

Le module nœud est en fait un composite. La (Figure III.7) montre la structure interne du module nœud. Les flèches pleines signifient le passage de message et les flèches en pointillés signifient une interface entre les modules et les appels de fonctions. Par exemple, la plupart des modules font appeler au gestionnaire de ressources pour signaler la consommation d'énergie.

Le Module d'application est le plus touché par l'utilisateur, le plus souvent par la création d'un nouveau module pour mettre en œuvre un nouvel algorithme. Les modules MAC, Routage, ainsi que le module de mobilité, sont également modifiés, généralement la création d'un nouveau module est en cause pour la création d'un nouveau protocole.

Tous les modules existants sont très accordés par de nombreux paramètres. En plus Castalia est basé sur OMNeT++ l'un des simulateurs les plus populaires pour les Réseaux de capteurs sans fils.

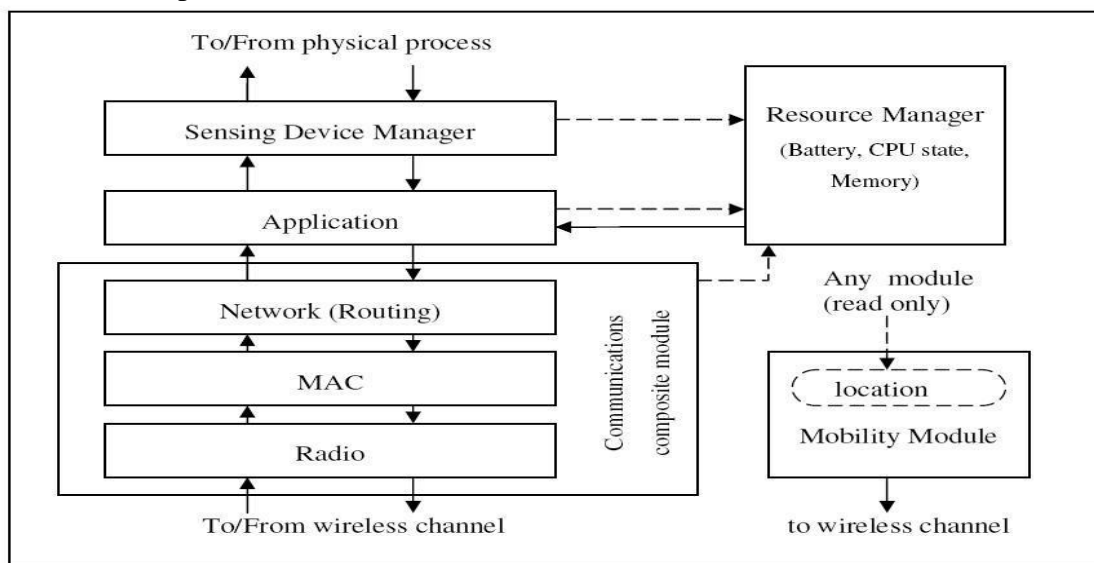


Figure 4.7 : Les principaux composants d'un nœud.

D'après ce que nous avons vu on peut conclure que castalia est le bon choix pour notre travail, à cause de son cadre "WSN" on peut facilement évaluer et ajouter des nouveaux protocoles.

4.8. Détails sur Castalia

Comme il est décrit précédemment Castalia est une plate forme sous OMNeT++ conçue particulièrement pour les RCSFs [49]. La structure de Castalia [50] est également reflétée dans l'hierarchie avec un code source. Chaque module contient un dossier "ned" qui définit le module lui même. Si le module est composé, les sous-répertoires existant définissent les sous-modules avec un code de C++ (cc, dossiers de .h) qui définit son comportement.

Cette hiérarchie complète des dossiers "ned" définit la structure globale du simulateur de Castalia.

Les principaux modules de Castalia sont :

- **Le module MAC :** La couche MAC joue un rôle très important en infectant les différents états aux nœuds pour améliorer l'efficacité énergétique : nœud en état de transmission (Transmit), d'écoute (Idle), ou en sommeil (Sleep). Le module "Radio" modifie ces états en fonctions des cas particuliers. Les utilisateurs peuvent modifier les paramètres aux exemples illustrés au niveau de Castalia pour la couche MAC en utilisant le fichier de configuration OMNeT++ (.ini). Par un simple réglage de certains paramètres des fichiers MAC "CSMA/CA", "TMAC", et "l'AMCC" de différentes simulations seront gérés.
- **Le module Radio :** Il est conçu sur la base des réseaux de capteurs sans fil avec trois états : le sommeil, la transmission et l'écoute. En outre, la consommation d'énergie sera différente dans chaque Etat. C'est la caractéristique principale de ce module. La figure à état fini peut donner une idée générale sur le changement d'état.
- **Canal sans fil :** Le module du canal sans fil est conçu pour modéliser l'environnement sans fil, qui est une évolution dynamique de l'environnement pour le nœud statique et mobile.
- **Le module Routage :** La fonction la plus importante du module de réseau est de recevoir le message de MAC module, et envoyer un message à la demande. Il dépend des informations d'entête.
- **Le module Application :** Le module Application est le module principal utilisé pour contrôler d'autres modules tels que le module de gestion des ressources, module de traitement physique, le module de routage, et le module radio. Ainsi ce module utilise également le mécanisme de message pour effectuer des actions différentes.

4.9. Outils de Simulation

4.9.1. Installation d'OMNET++ Version 4.6

Avant de commencer l'installation, il faut tout d'abord télécharger le code source omnetpp4.6 du site (<http://www.omnetpp.org/>) sans oublier de vérifier la version Linux, ensuite copier la source dans le dossier voulu et se positionner avec l'invite de commande dans ce dossier et suivre les étapes suivantes :

```
1. sudo tar xzf omnetpp-4.0b2-src.tgz -C /usr/local/ ;
2. cd /usr/local ;
3. sudo ln -s omnetpp-4.0b2 omnetpp ;
4. sudo apt-get install bison, flex, blt, lmodern, giftrans, doxygen,
libxml2-dev, graphviz, imagemagick ;
5. sudo apt-get install tcl8.4, tk8.4, tcl8.4-dev, tk8.4-dev ;
6. export LD_LIBRARY_PATH=$LD_LIBRARY_PATH :
/usr/local/omnetpp/lib ;
7. export TCL_LIBRARY=/usr/share/tcltk/tcl8.4 ;
8. export PATH=$PATH :/usr/local/omnetpp/bin ;
9. cd /usr/local/omnetpp ;
10. sudo ./configure ;
11. sudo make .
```

Figure 4.8 : Les étapes Installation d'OMNET++ Version 4.6.

Remarque : Les étapes 6, 7, 8 doivent être ajoutées dans le fichier "~/.bashrc".

4.9.2. Installation de Castalia 3.2

L'installation de castalia [50] est comme suit :

1. Récupérer le code source à partir du site <http://castalia.npc.nicta.com.au/>.
2. ouvrir un terminal et décompresser le fichier par la commande : `tar xvzf Castalia-3.2.tar.gz`
3. Un nouveau répertoire sera créé, nommé Castalia-3.2.
4. ajouter cette ligne `export PATH=$PATH : $HOME/Castalia-3.2/bin` à la fin du fichier `bashrc`.

5. taper la commande : `cd Castalia-3.2/`.
6. en suite `./makemake`.
7. `Make`

4.10. Implémentation d'un réseau

Castalia a pris en considération le développement de nouveaux protocoles au sein de cette plate forme. Dans ce contexte, les fichiers Template facilite la création d'un nouveau réseau avec les différentes couches.

Pour l'implémentation il suffit de copier les fichiers nécessaires dans un nouveau répertoire de votre choix. Nous avons adopté ce principe et créer un nouveau répertoire appelé "RCSF" ensuite copier les fichiers "yourNetwork.ned", "yourAppli-cation.ned", "yourHost.ned", "omnetpp.ini". Ces derniers contiennent des racines de procédures, où chacun peut modifié et élaboré son protocole.

Le programme sera complet que par l'ajout et la programmation des fichiers ".cc", ".h", ".msg" des différents modules. Une fois que tous les fichiers sont complets, la plateforme Castalia prendra en considération ce nouveau réseau seulement en exécutant les étapes de compilation suivante :

- a. Créer le fichier Makefile en exécutant " `opp_makemake -f`"
- b. Compiler en utilisant "make".

Après ces étapes, et seulement s'il n'existe pas d'erreurs un nouveau Fichier "yourNetwork" sera créer, en l'exécutant la simulation commencera.

4.11. Conclusion

Dans ce chapitre nous avons présentés les différents simulateurs existant pour les réseaux de capteur sans fil. Parmi ces simulateurs, notre choix a été fixé sur OMNET++ essentiellement à cause de sa construction modulaire et sa flexibilité. Le manque du simulateur OMNET++ au niveau des protocoles de réseaux de capteur à été solutionné avec l'implémentation des différentes plates formes tel que Castalia.

5.1 Introduction :

La simulation des réseaux de capteurs consiste principalement en la reproduction du comportement et du fonctionnement des nœuds capteurs dans un environnement informatique ; pour des raisons tel que : La répétition d'expérience, L'adressage des systèmes complexes, Le gain de temps et la variation des paramètres de simulation alors que la simulation réelle s'avère coûteuse, voir impossible dans quelque cas.

Le développement d'un nouveau protocole MAC pour le réseau de capteur sans fil nécessite une phase de simulation avant leur implémentation réelle. Dans ce chapitre nous intéressons aux protocoles TMAC pour les RCSFs.

Notre travail consiste à simuler le protocole TMAC et démontrer la performance et l'efficacité de ce protocole en termes d'économie d'énergie, en utilisant la plateforme Castalia- 3.2 avec le simulateur OMNET++.

5.2 Environnement de travail :

Nous allons détaillés les outils utilisés dans la réalisation de notre simulation.

- Modèle : TOSHIBA.
- Disque dur : 160 Go.
- Processeur : Core2Duo 2.20 GHz
- Mémoire (RAM) : 4Go.
- ✓ Système d'exploitation : Linux Distribution Mint.
- ✓ Le simulateur OMNet++ 4.6.
- ✓ Le Simulateur Castalia 3.2.

5.3 Réalisation :

5.3.1 Objectif :

- Notre objectif est de simuler un protocole MAC au niveau de la plateforme Castalia, et ainsi de tester ce protocole.
- Démonstration des performances de protocole TMAC en termes d'économie d'énergie.
- La comparaison entre le protocole TMAC par rapport aux des périodes (veille, écoute) et écoute seulement et en différentes nombres des nœuds.

5.4 Description détaillé :

A. Description de la conception d'un nœud :

Un réseau de capteur sans fil est caractérisé par le déploiement aléatoire des nœuds. Pour notre simulation, la 1ere chose qu'on devait modéliser est le module nœud avec ses différents sous modules (La pile protocolaire). Pour une vue mieux explicatif, le fichier graphique ".Ned" illustre les différents modules utilisés.

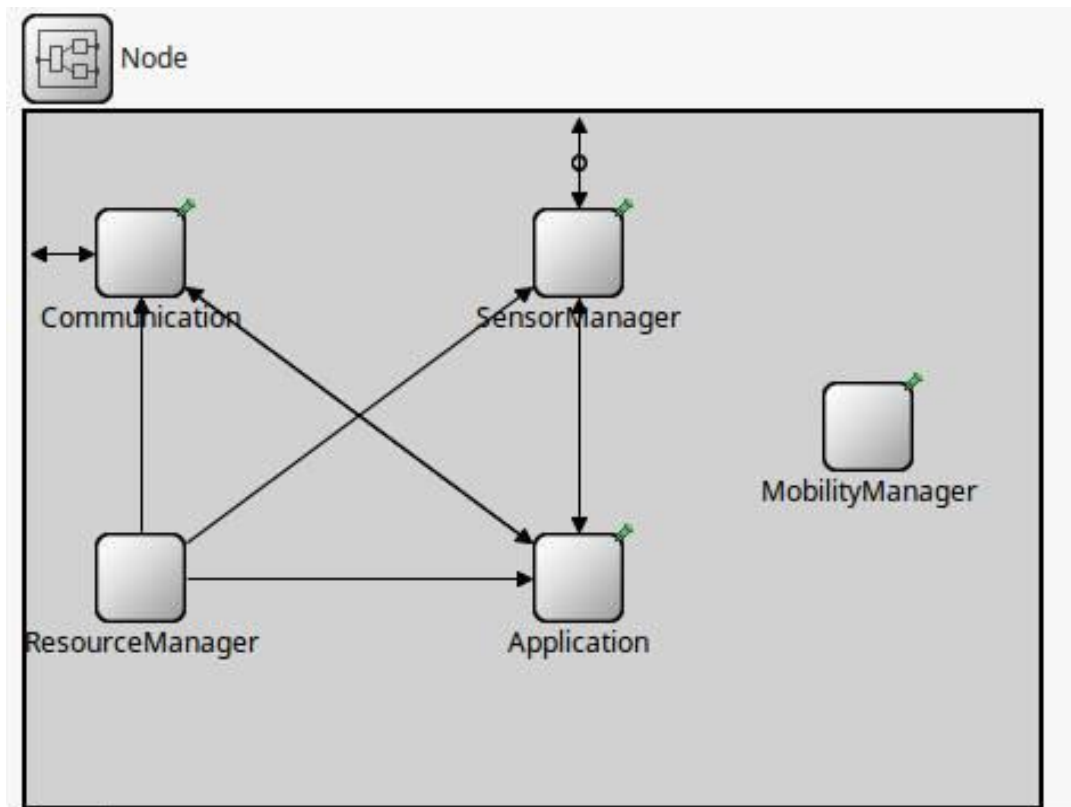


Figure 5.1 : Fichier graphique ".ned" du nœud.

B. Description du réseau :

La structure du réseau est défini dans le fichier " SensorNetwork.ned " (voir figure 5.2). Un réseau de capteur sans fil est composé du module nœud décrit précédemment ainsi d'un canal pour la communication entre nœud et processus physique pour la gestion des données capturées par les capteurs.

Les connexions entre nœuds sont gérées au niveau de cette structure dans "Connections" mais nous avons choisis de laisser une génération aléatoire pour se rapprocher plus aux concepts des Réseaux De Capteurs Sans Fil.

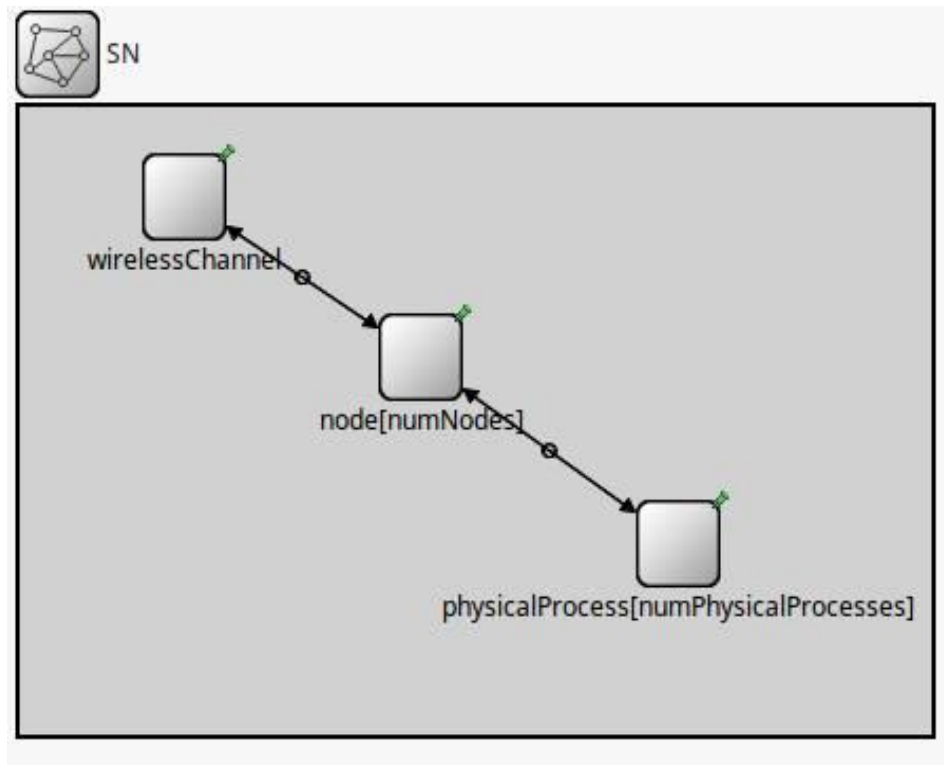


Figure 5.2 : Le fichier SensorNetwork.ned.

C. Paramètres de simulation :

La surface de déploiement et la durée de simulation se varie selon le nombre des nœuds.

Le détail des paramètres de simulation est présenté dans le tableau suivant :

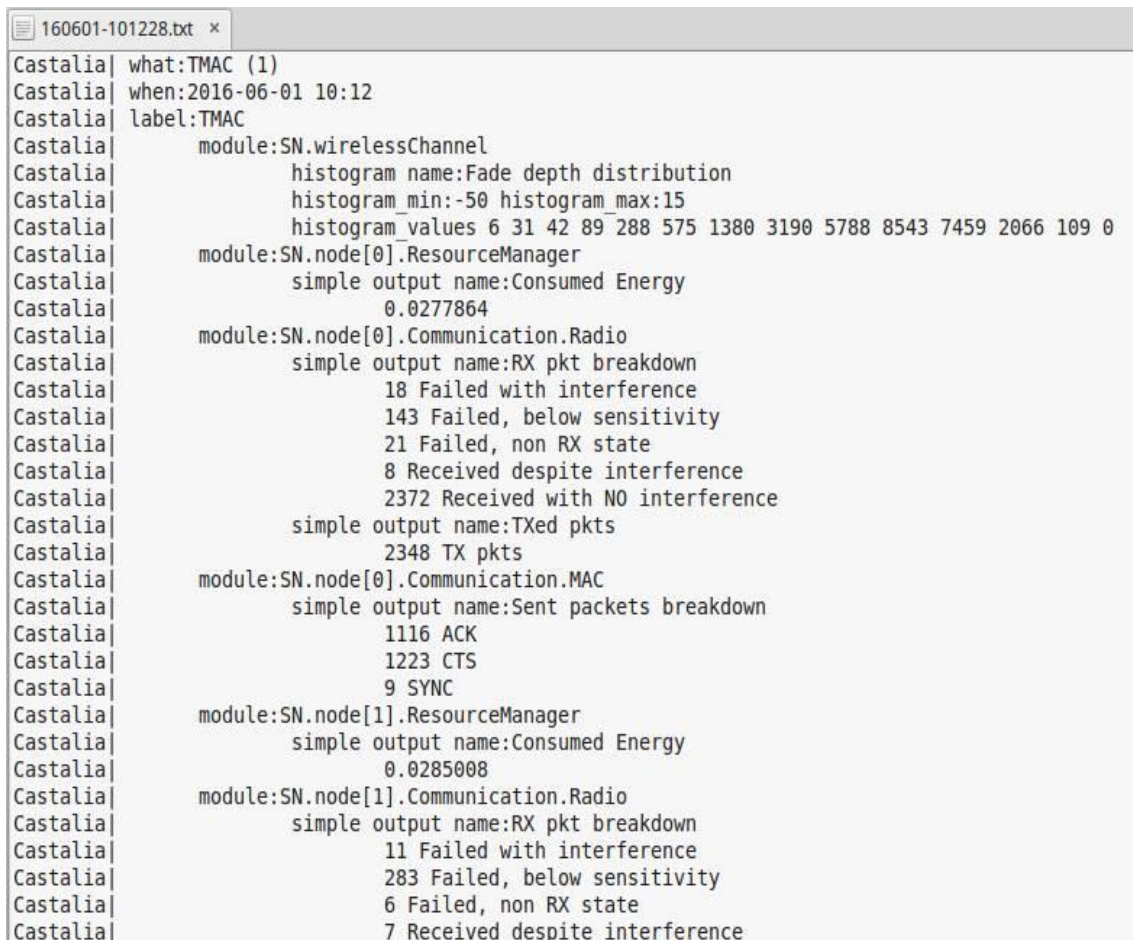
Temps de simulation	Varie selon le nombre des noeuds
Fichier des Paramètres radio	BANRadio
symboleRSSI	16
Puissance de Transmission	-15dBm
Application	throughputTest
Energie initiale	18720 joules
Temps d'écoute	61 ms
Longueur de frame	610 ms
Taille des paquets RTS/CTS	13 bytes
Taille du paquet ACK	11 bytes
Taille du paquet SYNC	11 bytes
Période de contention	10 ms
Temps de RESYNC	6

Tableau 5.1 : Les paramètres utilisés pour la simulation de protocole TMAC.

5.5 Résultats et analyses de simulation de T-MAC :

À la fin de l'exécution la génération de deux fichiers se créer.

- ✚ **Fichier trace :** donne une vue générale et détaillé du processus parcouru par chaque nœud durant la simulation.
- ✚ **Fichier résultat :** affiche un résumé sur les informations collectées à la fin de l'exécution comme le nombre de paquets reçus, les paquets perdus, et la consommation d'énergie. Par la suite, on montre ces résultats sous forme des graphes.



```

160601-101228.txt x
Castalia| what:TMAC (1)
Castalia| when:2016-06-01 10:12
Castalia| label:TMAC
Castalia|   module:SN.wirelessChannel
Castalia|           histogram name:Fade depth distribution
Castalia|           histogram_min:-50 histogram_max:15
Castalia|           histogram_values 6 31 42 89 288 575 1380 3190 5788 8543 7459 2066 109 0
Castalia|   module:SN.node[0].ResourceManager
Castalia|       simple output name:Consumed Energy
Castalia|           0.0277864
Castalia|   module:SN.node[0].Communication.Radio
Castalia|       simple output name:RX pkt breakdown
Castalia|           18 Failed with interference
Castalia|           143 Failed, below sensitivity
Castalia|           21 Failed, non RX state
Castalia|           8 Received despite interference
Castalia|           2372 Received with NO interference
Castalia|       simple output name:TXed pkts
Castalia|           2348 TX pkts
Castalia|   module:SN.node[0].Communication.MAC
Castalia|       simple output name:Sent packets breakdown
Castalia|           1116 ACK
Castalia|           1223 CTS
Castalia|           9 SYNC
Castalia|   module:SN.node[1].ResourceManager
Castalia|       simple output name:Consumed Energy
Castalia|           0.0285008
Castalia|   module:SN.node[1].Communication.Radio
Castalia|       simple output name:RX pkt breakdown
Castalia|           11 Failed with interference
Castalia|           283 Failed, below sensitivity
Castalia|           6 Failed, non RX state
Castalia|           7 Received despite interference

```

Figure 5.3 : Fichier résultat.txt du Castalia

La simulation est établie sur un champ de 100m x 100m avec un nombre de 5 nœuds déployés d'une manière uniforme. Les résultats de la simulation sont enregistrés dans le fichier résultat sous Castalia.


```

omnetpp.ini  Castalia-Trace.txt
1.87988881947 SN.node[3].Communication.MAC Resetting MAC state to default, reason: sensed carrier
1.87988881947 SN.node[3].Communication.MAC Delivering [CS Interrupt] to Network layer
1.87988881947 SN.node[2].Communication.MAC Resetting MAC state to default, reason: sensed carrier
1.87988881947 SN.node[2].Communication.MAC Delivering [CS Interrupt] to Network layer
1.87988881947 SN.node[0].Communication.MAC Resetting MAC state to default, reason: sensed carrier
1.87988881947 SN.node[0].Communication.MAC Delivering [CS Interrupt] to Network layer
1.879987726623 SN.node[1].Communication.MAC Resetting MAC state to default, reason: timeout
1.880997863415 SN.node[5].Communication.MAC send RTS packet5
1.881031769664 SN.node[4].Communication.MAC Resetting MAC state to default, reason: sensed carrier
1.881031769664 SN.node[4].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881031769664 SN.node[0].Communication.MAC Resetting MAC state to default, reason: sensed carrier
1.881031769664 SN.node[0].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881031769664 SN.node[1].Communication.MAC Resetting MAC state to default, reason: sensed carrier
1.881031769664 SN.node[1].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881035675914 SN.node[2].Communication.MAC Resetting MAC state to default, reason: sensed carrier
1.881035675914 SN.node[2].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881176300914 SN.node[0].Communication.MAC send CTS packet5
1.881210207163 SN.node[4].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881210207163 SN.node[2].Communication.MAC Resetting MAC state to default, reason: sensed carrier
1.881210207163 SN.node[2].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881210207163 SN.node[1].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881354738413 SN.node[5].Communication.MAC send DATA packet5
1.881388644662 SN.node[4].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881388644662 SN.node[1].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881392550912 SN.node[2].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881431613412 SN.node[0].Communication.MAC send ACK packet5
1.881465519661 SN.node[4].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881465519661 SN.node[2].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881465519661 SN.node[1].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881564433878 SN.node[0].Communication.MAC Delivering [CS Interrupt] to Network layer
1.881594425911 SN.node[5].Communication.MAC Resetting MAC state to default, reason: timeout
1.881594425911 SN.node[5].Communication.MAC Transmission succesful to 0
1.881682865643 SN.node[5].Communication.MAC Resetting MAC state to default, reason: transmission succ
1.881682865643 SN.node[1].Communication.MAC Resetting MAC state to default, reason: timeout

```

Figure 5.4 : Fichier trace.txt du Castalia

Cinq nœuds déployés d'une manière uniforme sont suffisantes pour couvrir l'overhearing, l'idle listening, les collisions et la synchronisation. Les résultats obtenus pour le protocole TMAC sont présentés graphiquement dans les figures ci-dessus :

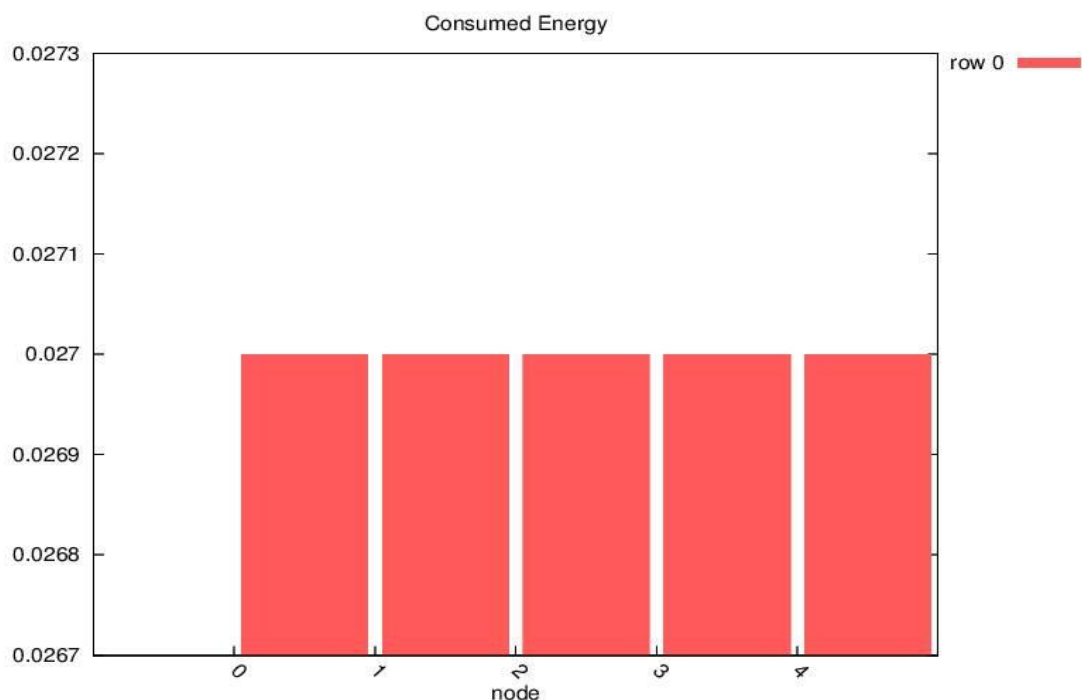


Figure 5.5 : L'énergie consommée par chaque nœud

La figure 5.5 illustre L'énergie consommée pour tous les cinq nœuds sans fil réseau de capteurs. On remarque que tout le nœud de même consommation d'énergie.

TMAC suit horaire de sommeil dynamique. Le protocole TMAC introduit un mécanisme de délai d'attente active qui diminue l'écoute inactif charge d'overhead en ajustant dynamiquement la période active selon le trafic réseau.

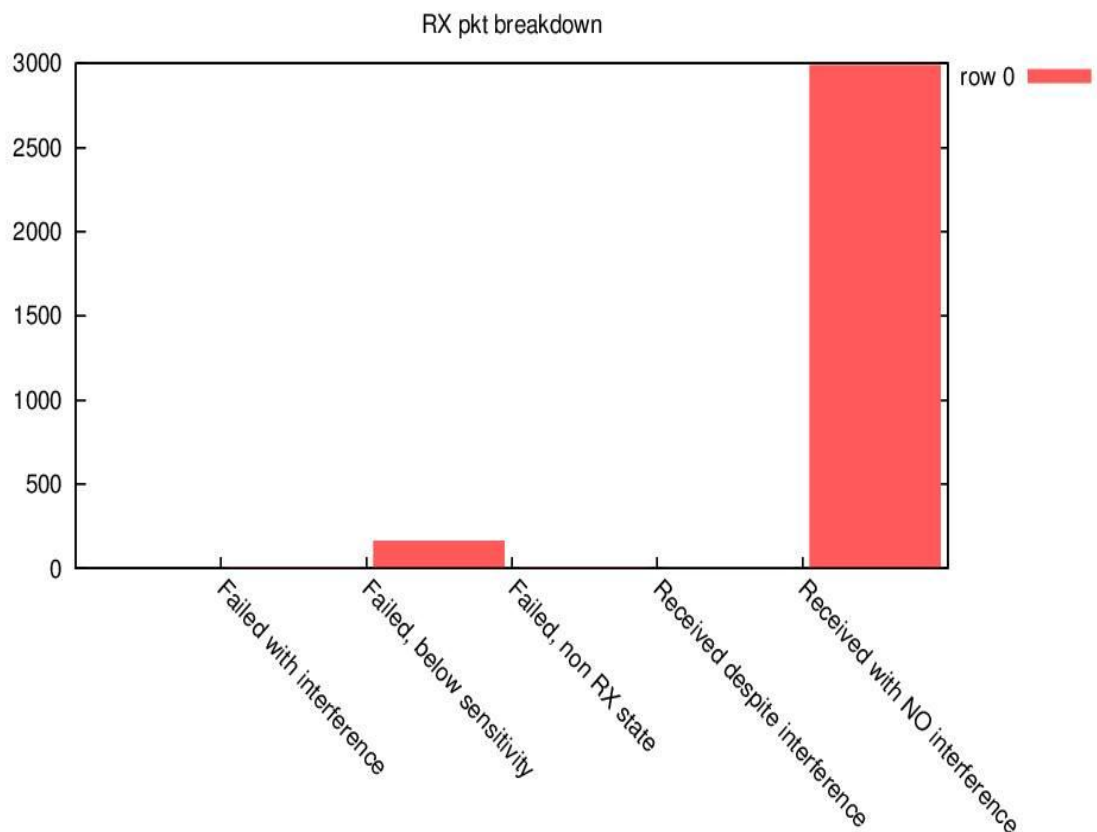


Figure 5.6 : détail des paquets reçus.

La figure 5.5 illustre en détail des paquets reçus pour tous les cinq nœuds sans fil réseau de capteurs. la figure montre que le TMAC a réduit considérablement le nombre des paquets reçu.

La période « active » se termine quand il n'y a plus aucun événement de réception de message pendant une durée TA (Timeout Activity) qui détermine la durée d'écoute par séquence actif/endormi. Par conséquent, la période active est adaptée à la charge du trafic.

○ Comparaison :

Les deux graphes représentés dans les figures : 5.7, 5.8 représentent l'énergie moyenne consommée et détail des paquets reçus par plusieurs simulations : TMAC ; TMAC 20 ; TMAC 60 ; TMAC 100.

Le protocole de T-MAC se sert de la (RTS/CTS) Mécanisme d'allocation de canal pour fins d'évitement de collision et pour lutter contre la présence éventuelle de nœuds cachés.

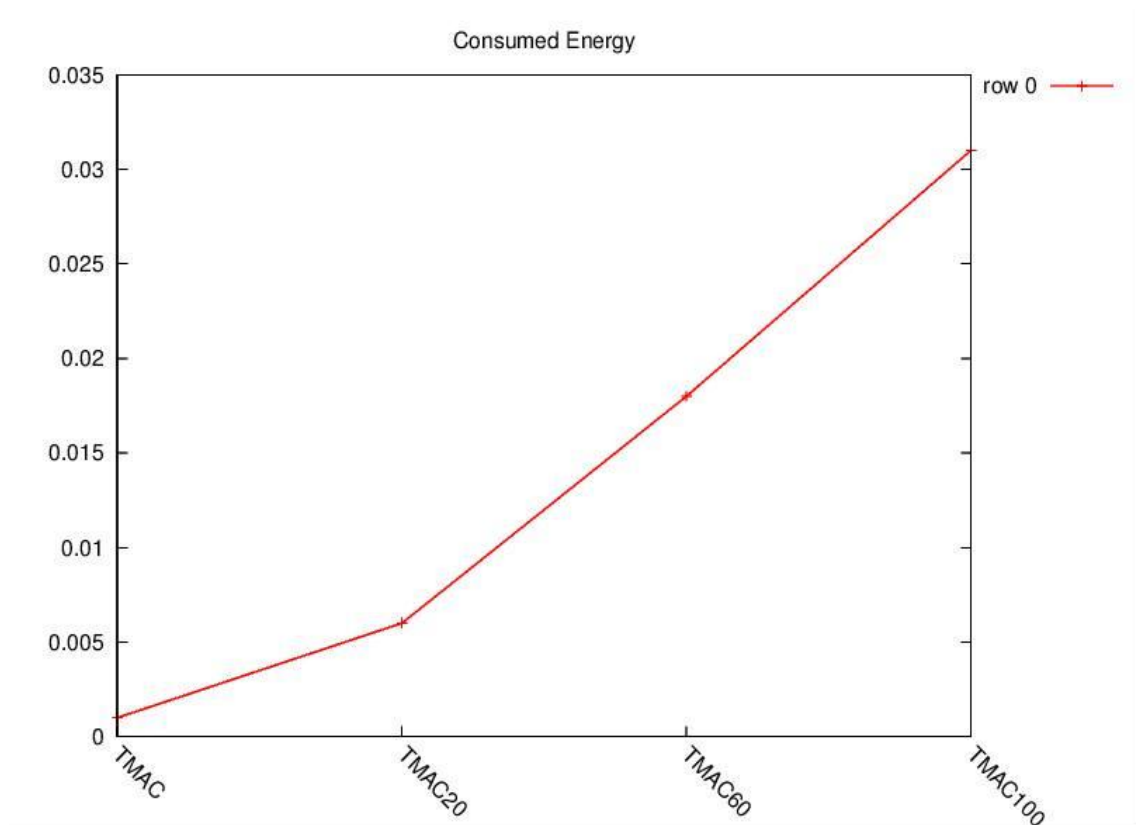


Figure 5.7 : L'énergie moyenne consommée par plusieurs simulations.

Les nœuds sur la limite comparativement moins énergivores. Cette expérience montre l'énergie meilleure efficacité de T Mac Protocol. Efficacité énergétique a été la question de la conception principale pour le développement du protocole TMAC.

Nous pouvons remarquer que l'énergie moyenne consommée dans plusieurs simulations est augmentée quand le nombre des nœuds augmente.

Les simulations ont montré que le protocole de T-MAC introduit un moyen de diminuer l'énergie consommation dans un environnement instable, où le taux de messages fluctue, soit en temps ou en location.

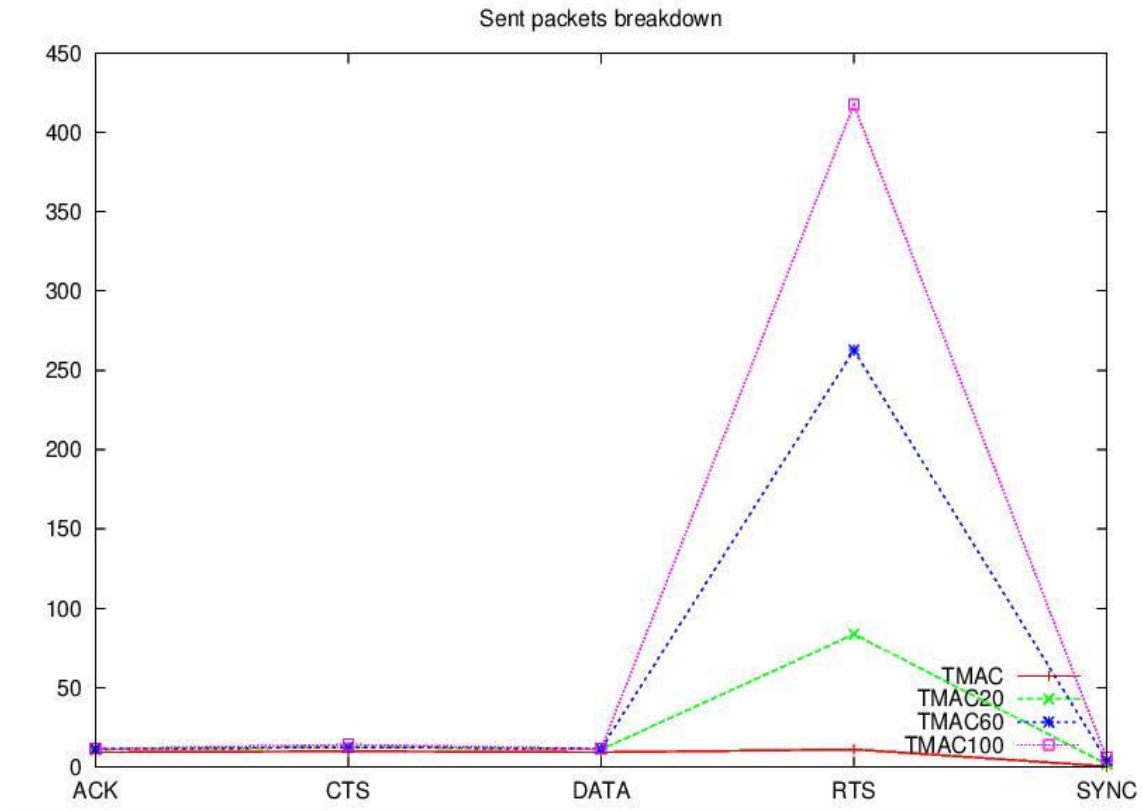


Figure 5.8 : Nombre de paquets transmis par plusieurs simulations.

T Mac Protocol est plus éconergétique en raison de l'introduction de la temporisation d'activation. FRTS (requête avenir d'envoi) est également être ajouté à T Mac Protocol, qui est responsable du grand nombre de paquets de contrôle en T Mac Protocol.

5.6 Conclusion

Des études antérieures ont montré que T Mac est mieux mac Protocole parce que les principaux critères de performances dans les réseaux de capteurs sans fil est énergie réseau et la consommation à vie. Cette étude donne un plus vue détaillée de ces protocoles. Nous avons proposé le protocole de T-MAC pour résoudre le problème d'écoute inactif dans un réseau de capteurs sans fil. T-MAC adapte dynamiquement un cycle Ecoute/veille d'une nouvelle façon, à travers délais de grains fins, tout en ayant la complexité minimale.

Conclusion générale

Les réseaux de capteurs sans fil ont été maintenus en évolution en raison des progrès réalisés dans diverses technologies telles que la radio, systèmes de batteries et d'exploitation dans les éléments de capteur, mais mac protocoles sont encore plus importants dans RCFS parce que l'exacte mise en œuvre de la communication entre les capteurs est dérivée par les protocoles de mac.

La consommation de la batterie, la durée de vie du réseau, latence de communication, des collisions de paquets sont parfois très facteurs importants ceux dépend des protocoles de mac utilisés dans un les réseaux de capteurs sans fil. Dans ce mémoire nous avons proposé le protocole T-MAC.

T-MAC adapte dynamiquement une écoute cyclique / sommeil d'une manière nouvelle, par les délais d'attente à grains fins, tout en ayant une complexité minimale. Des simulations ont montré que le protocole T-MAC introduit un moyen de réduire la consommation d'énergie un environnement volatile où le taux de message fluctue, soit dans le temps ou dans un endroit. Nous avons également identifié un problème avec le protocole T-MAC (Le problème de sommeil tôt) et proposé des solutions originales pour ce problème (FRTS et priorité-tampon pleine).

DIFS : Distributed Coordination Function InterFrame Spacing.

SIFS : Short InterFrame Spacing.

SMAC : Sensor-MAC.

TMAC : Timeout-MAC.

FRTS : Future Request To Send.

FDMA : Frequency Division Multiple Access.

RCSF : Réseau de capteurs sans fil.

WLAN : Wireless Local Area Network.

WMAN : Wireless Metropolitan Area Network.

WPAN : Wireless Personal Area Network.

WWAN : Wireless Wide Area Network.

TA : Timeout Activation.

D-MAC : Data gathering MAC.

LPL : Low Power Listening.

B-MAC : Berkeley-MAC.

DW-LPL : Dual Wake-up LPL.

Z-MAC : Zebra-MAC.

G-MAC : Gateway MAC.

GTIM : Gateway Traffic Indication Message.

SYNC : Synchronization.

NAV : Network Allocation Vector.

OMNeT++ : Objective Modular Network Testbed in C++.

GNU : General Public License.

UDP : User Datagram Protocol.

TCP : Transmission Control Protocol.

FIFO : First In First Out.

NS : Network Simulator.

OTCL : Object Oriented Tool Command Language.

GloMoSim : Global Mobile information système Simulateur.

OSI : Open Systems Interconnection.

UCLA : Parallel Computing Laboratory.

PARSEC : PARallel Simulation Envirennement for Complex system.

IDE : Integrated Developement Envirenement.

XP : eXPerience -NT4.0 : New Technology.

Gcc : GNU Compiler Collection.

CPU : Central Processing Unit.

BAN : Body Area Networks.

WSN : Wireless Sensor Network.

NICTA : National ICT Australia.

MinGW : Minimalist GNU for Windows.

JDK : Java Developement Kit.

MEMS : Microelectromechanical Systems.

MAC : Médium Access Control.

LAN : Local Area Network.

GSM : Global System for Mobile Communications.

GPS : Global Positionning System.

TDMA : Time Division Multiple Access.

CDMA : Code Division Multiple Access.

CSMA : Carrier Sense Multiple Access.

CSMA/CD : Carrier Sense Multiple Access /Collision Detect.

IEEE : Institute of Electrical and Electronics Engineers.

CSMA/CA : Code Division Multiple Access / Collision Avoidance.

TRAMA : TRaffic-Adaptive Medium Access Control.

NP : Neighbor Protocol. - SEP : Schedule Exchange Protocol.

AEA : Adaptative Election Algorithm.

FLAMA : FLOW-Aware Medium Access.

CTS : Clear to Send.

RTS : Request to Send.

ACK : Acquittement.

DCF : Distributed Coordination Function.

PCF : Point Coordination Function.

CCA : Clear Channel Assessment.

- [01] Samia Maarouf, Souhila Ouadah. Implémentation et évaluation des schémas de routage sur une plateforme réelle de réseaux de capteurs sans fil. (R.S.D). Université Tlemcen 2014.
- [02] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "A Survey on Sensor Networks", IEEE Communications Magazine, Vol.40, No. 8, pp. 102-114, 2002.
- [03] J. P. Lynch, A. Sundararajan, K. H. Law, A. S. Kiremidjian, E. Carrier, "Embedding damage detection algorithms in a wireless sensing unit for operational power efficiency", Smart Materials and Structures, Vol. 13, No. 4, pp. 800-810, 2004.
- [04] H. Karl, A. Willing, "Protocols and Architectures for Wireless Sensor Networks", Book published by Wiley & Sons, 2007.
- [05] Gaurav Jolly, Mustafa C. Kusçu, Pallavi Kokate et Mohamed Younis, "A Low-Energy Key Management Protocol for Wireless Sensor Networks", (ISCC'03), IEEE COMPUTER SOCIETY, 2003
- [06]. H. Baldus, K. Klabunde, G. Muesch, "Reliable set-up of medical body-sensor networks". in EWSN 2004. January 2004.
- [07]. E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, V.Z. Groza. "Sensor-based information appliances". IEEE Instrumentation Measurement Magazine. Vol. 3 (4), pp. 31-35. December 2000.
- [08] :KAZI TANI Chahrazad. BENHADDOUCHE Wiam. Implémentation et test d'un protocole de prévention de l'attaque Clone dans un réseau de capteurs sans fil. Mémoire de fin d'études Pour l'obtention du diplôme de Master en Informatique Option: Réseaux et systèmes distribués (R.S.D). Université Abou Bakr Belkaid-Tlemcen Algérie. 2014.
- [09]. Akyildiz, I.F., W. Su, Y. Sankarasubramaniam, E. Cayirci, "A Survey on Sensor Networks", IEEE Communications Magazine, August, 102-114(2002).
- [10]. I.F. Akyildiz, W.S. Sankarasubramaniam, E. Cayirci, "Wireless Sensor Network: A Survey". Computer networks, Vol. 38, pp. 393-422. 2002.
- [11] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Micro sensor Networks, In proc of the Hawaii International Conference on Systems Science, vol. 8, pp. 8020, January 2000.

- [12] : Cedric RAMASSAMY, « Analyse des protocoles des Réseaux de capteurs sans-fil », en vue de l'obtention du DOCTORAT DE L'UNIVERSITE DES ANTILLES ET DE LA GUYANE Spécialité : Informatique, le 23 Novembre 2012 au D'département Scientifique Inter facultaire Campus.
- [13] Demirkol I., Ersoy C., Alagöz F., « MAC Protocols for Wireless Sensor Networks: a Survey », 2003.
- [14] K. Akkaya et M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, Vol.3, no. 3, pp. 325–349, 2005.
- [15] C. Intanagonwiwat, R. Govindan et D. Estrin, "Directed Diffusion: a scalable et robust communication paradigm for sensor networks," *ACM Press*, 2000.
- [16] A.A. Ahmed, H. Shi, et Y. Shang, "A Survey on Network Protocols for Wireless Sensor Networks," *Proc. IEEE Int Conf. Information Technology: Research et Education (ITRE '03)*, 2003.
- [17] J. Kulik, W.R. Heinzelman, H. Balakrishnan, "Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks," In: *Wireless Networks*, Vol. 8, pp. 169-185, 2002.
- [18] Y. Xu, J. Heidemann, et D. Estrin, "Geography-informed energy conservation for ad hoc routing," *7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 01)*, Rome, Italy, July 2001.
- [19] S. Bandyopadhyay, E. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," *Proceedings of IEEE INFOCOM*, Vol. 3, pp. 1713-1723, 2003.
- [20]: W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", *33rd Hawaii International Conference on System Sciences*, Hawaii, USA, pp. 3005-3014, Janvier 2000.
- [21]: Boubiche Djallel Eddine, Une approche Inter-Couches (cross-layer) pour la Sécurité dans les R.C.S.F, En vue de l'obtention du diplôme de Doctorat en Sciences en Informatique, Université de Batna

- [22] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in IEEE Infocom, 2002, pp. 1567–1576.
- [23] J. Polastre, J. Hill, et D. Culler, “Versatile low power media access for wireless sensor networks,” in ACM Sensys, Nov 2004.
- [29] International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6, January 2013.
- [30] Rajesh Yadav, Shirshu Varma, N. Malaviya, “A SURVEY OF MAC PROTOCOLS FOR WIRELESS SENSOR NETWORKS”, UbiCC Journal, Volume 4, Number 3, August 2009.
- [31] Sarika Khatarkar et.al / Indian Journal of Computer Science and Engineering (IJCSE) ISSN : 0976-5166 Vol. 4 No.4 Aug-Sep 2013 304.
- [32] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In 21st Conference of the IEEE Computer and Communications Societies (INFOCOM), volume 3, pages 1567–1576, June 2002.
- [33] LAN MAN Standards Committee of the IEEE Computer Society. IEEE Std 802.11-1999, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE, 1999.
- [34] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys 2003, pages 171–180, Los Angeles, California, USA, November 2003. ACM.
- [35] Van Dam, Langendoen K., « An Adaptive EnergyEfficient MAC Protocol for Wireless Sensor Networks », Faculty of Information Technology and Systems, Delft University of Technology, Pays-Bas, 2003.
- [36] Heidemann J., Ye W., Estrin D., « An Energy-Efficient MAC Protocol for Wireless Sensor Networks », IEEE Infocom, Juin 2002.
- [37] Julien Vaudour, Vincent Gauthier Rapport de Recherche INT N°06006RST GET/INT;UMR5157 SAMOVAR Institut National des T´el´ecommunications, Evry, France.

- [38] : Leila Imane NIAR, Analyse Graphique pour la surveillance dans un réseau de capteurs sans fils (RCSF) Simulateur : OMNET++, Mémoire Magister : Informatique, JUILLET 2012.
- [39] Bilge Cetin, "SIMULATION ENVIRONMENT", Novembre 2005
- [40] [http ://www.isi.edu/nsnam/ns/](http://www.isi.edu/nsnam/ns/)
- [41] Abdallah Makhoul, "Réseaux de capteurs : localisation, couverture et fusion de données.", THÈSE pour obtenir le grade de DOCTEUR de l'Université de Franche-Comté 2008.
- [42] C. Mallanda, A. Suri, V. Kunchakarra, S.S. Iyengar*, R. Kannan* and A. Durresi "Simulating Wireless Sensor Networks with OMNeT++", S. Sastry The University of Akron, Akron, Ohio.
- [43] : Rachid Bechar, "Théorie de la redondance pour la reconfiguration des systèmes - Application aux réseaux de capteur sans fil", Mémoire de magister Ecole doctorale STIC, Université Abdelhamid Ibn Badis de Mostaganem. Département d'informatique, 2009.
- [44] "OMNeT++, Discrete Event Simulation System Version 4.0", User Manual.
- [45] " A Quick OverView of the OMNeT++4.0 IDE"
- [46] " OMNeT++ Installation Guide Version 4.1", 2010 P,Q :
- [47] Azadeh Abdolrazaghi, "Unifying Wireless Sensor Network Simulators", Master's Degree Project Stockholm, Sweden, 2009.
- [48] LIANG LI, " Implementing Transport Protocol PSFQ in Sensor Network in Castalia", School of Computer Science, 2009.
- [49] : Maya Sayad "Energy Efficient Protocol (EEP) : un protocole de routage efficace en énergie pour réseaux de capteurs sans fil", Mémoire de fin d'études Pour l'obtention du diplôme d'ingénieur d'état en informatique ; 2008/2009.
- [50] : A. Boulis, "Castalia, a simulator for wireless sensor networks and body area networks, version 3.1", User's manual, NICTA, December 2010.

Résumé : Les réseaux de capteurs sans fil ont été maintenu en évolution en raison des progrès de diverses technologies comme la radio, batterie et systèmes d'exploitation en éléments de capteur mais les protocoles MAC sont encore plus importants dans les WSN parce que l'application exacte de communication entre les capteurs est dérivée par les protocoles MAC. Consommation de la batterie, durée de vie de réseau, latence de communication, paquet de collisions sont très importantes facteurs ceux dépend des protocoles MAC utilisé dans un réseau de capteurs sans fil. T Mac a été le protocole de point de repère dans les protocoles de réseaux de capteurs sans fil en raison de leur utilité et la facilité d'implémentation avec simplicité. Pour l'étude de TMac, nous avons utilisé Castalia. Castalia utilise Les caractéristiques d'OMNeT et est conçu en particulier pour les réseaux de capteurs sans fil.

Mots clés : TMac, protocole, Castalia, Omnetpp, WSN.

Abstract: Wireless sensor networks have been kept evolving due to the advancements in various technologies like radio, battery and operating systems in sensor elements but MAC protocols are still most important in WSN because the exact implementation of communication among sensors is derived by the MAC protocols. Battery consumption, network lifetime, communication latency, packet collisions are some very important factors those depends on MAC protocols used in a wireless sensor networks. T Mac has been landmark protocol in wireless sensor networks protocols because of their utility and ease of implementation along with simplicity. For the study of TMac, we used Castalia. Castalia uses the OMNT characteristics and is designed especially for wireless sensor networks.

Keywords: T Mac, protocol, Castalia, Omnetpp, WSN.

المخلص : شبكات الاستشعار اللاسلكية ظلت تتطور بسبب التقدم في مختلف التكنولوجيات مثل الإذاعة، البطارية وأنظمة التشغيل في عناصر أجهزة الاستشعار ولكن بروتوكولات MAC ما زالت مهمة جداً في WSN نظراً للتنفيذ الدقيق للاتصال بين أجهزة الاستشعار من خلال بروتوكولات MAC. استهلاك البطارية، عمر شبكة، كمون الاتصال، حزمة الاصطدامات مهمة جداً في بعض العوامل التي تعتمد على بروتوكولات MAC المستخدمة في شبكات الاستشعار اللاسلكية. TMac من أهم بروتوكولات شبكات استشعار لاسلكية بسبب فائدته وسهولة التنفيذ جنباً إلى جنب مع البساطة. لدراسة TMac، استخدمنا Castalia. Castalia يستخدم خصائص Omnetpp، وهي مصممة خصيصاً لشبكات استشعار لاسلكية.

الكلمات الرئيسية: TMac, protocole, Castalia, Omnetpp, WSN.