

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaïa

Faculté des Sciences Exactes

Département d'Informatique



*Mémoire de Master Professionnel*

en Informatique

**Option**

Administration et Sécurité des Réseaux

**Thème**

---

**Simulation d'un réseau de capteur sous Cooja avec  
adaptation du protocole CSMA/CA au temps réel**

---

M. *BADJADJ* Youcef

Présenté par :

M. *MECHRAFI* M<sup>d</sup> Cherif

Encadré par :

M. *ACHROUFENE* Achour

Soutenu devant le jury composé de :

Présidente M<sup>me</sup> ALOUI Sorya

Examinatrice M<sup>elle</sup> BOUADEM Nassima

Examinatrice M<sup>elle</sup> CHEKLAT Lamia

Promotion 2014/2015

# TABLE DES MATIÈRES

Table des Matières	i
Table des figures	vi
Liste des tableaux	v
Liste des abréviations	vi
Introduction Générale	1
<b>1 Introduction aux réseaux de capteurs</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Réseaux informatiques . . . . .	3
1.2.1 Définition d'un réseau . . . . .	3
1.2.2 Intérêt d'un réseau . . . . .	4
1.2.3 Topologie d'un réseau . . . . .	4
1.2.3.1 Topologie logique . . . . .	4
1.2.3.2 Topologie physique . . . . .	4
1.2.4 Classification des réseaux . . . . .	5
1.2.5 Architecture des réseaux . . . . .	6
1.2.6 Modèles de communication OSI . . . . .	7
1.2.7 Type des réseaux . . . . .	8
1.2.7.1 Réseaux filaires . . . . .	8
1.2.7.2 Réseaux sans fil . . . . .	8
1.3 Réseaux de capteur sans fil . . . . .	8
1.3.1 Historique des réseaux de capteurs sans fil . . . . .	8
1.3.2 présentation des réseaux de capteurs . . . . .	9

1.3.2.1	Définition d'un capteur . . . . .	9
1.3.2.2	Architecture d'un nœud (capteur) . . . . .	9
1.3.3	Définition d'un RCSF ou WSN (Wireless Sensor Network) . . . . .	10
1.3.4	Architecture de communication dans les réseaux de capteurs . . . . .	11
1.3.5	Caractéristiques d'un réseau de capteur sans fils . . . . .	12
1.3.6	Topologies des réseaux de capteurs . . . . .	13
1.3.7	Systèmes d'exploitation pour les réseaux de capteurs . . . . .	14
1.3.7.1	Définition d'un système d'exploitation . . . . .	14
1.3.7.2	Caractéristiques des systèmes d'exploitation pour RCSF . . . . .	14
1.3.7.3	Quelque systèmes d'exploitation existants . . . . .	15
1.3.7.4	Comparaison entre ces systèmes d'exploitation . . . . .	17
1.3.8	Domaines d'application des réseaux de capteurs . . . . .	18
1.3.8.1	Domaine militaire . . . . .	18
1.3.8.2	Domaine commerciale . . . . .	18
1.3.8.3	Domaine environnementale . . . . .	18
1.3.8.4	Domaine médical . . . . .	18
1.4	Conclusion . . . . .	19
<b>2</b>	<b>La couche MAC dans IEEE.802.15.4</b>	<b>20</b>
2.1	Introduction . . . . .	20
2.2	Normes IEEE 802 . . . . .	20
2.2.1	Standard IEEE . . . . .	20
2.2.2	Normes IEEE 802.x . . . . .	21
2.2.2.1	Norme 802.3 . . . . .	21
2.2.2.2	Norme 802.11 . . . . .	22
2.2.2.3	Norme 802.12 . . . . .	22
2.2.2.4	Norme 802.15 . . . . .	22
2.3	IEEE 802.15.4 . . . . .	24
2.3.1	Modes de fonctionnement dans IEEE 802.15.4 . . . . .	25
2.3.2	Architecture en couches . . . . .	26
2.3.2.1	Couche Physique dans IEEE 802.15.4 . . . . .	26
2.3.2.2	Couche MAC dans IEEE 802.15.4 . . . . .	26
2.4	CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance) . . . . .	27
2.4.1	Déroulement du protocole CSMA/CA . . . . .	27
2.4.2	Organigramme de la procédure CSMA/CA . . . . .	29
2.5	Méthodes d'accès au canal . . . . .	30
2.6	Conclusion . . . . .	33

<b>3</b>	<b>Adaptation de CSMA/CA pour le temps réel</b>	<b>34</b>
3.1	Introduction . . . . .	34
3.2	Problématique . . . . .	34
3.3	Objectif de notre travail . . . . .	35
3.4	Contiki en détails . . . . .	36
3.4.1	Fonctionnement de CSMA/CA dans Contiki . . . . .	38
3.4.2	Quelques limitations temps réel du CSMA dans Contiki . . . . .	39
3.5	CSMA adapté . . . . .	40
3.5.1	Fonctionnement . . . . .	41
3.6	Conclusion . . . . .	42
<b>4</b>	<b>Simulation</b>	<b>43</b>
4.1	Introduction . . . . .	43
4.2	Simulateur Cooja . . . . .	43
4.3	Déroulement des tests . . . . .	44
4.4	Csma-contiki . . . . .	45
4.4.1	Analyse des résultats . . . . .	45
4.4.2	Interprétation des résultats . . . . .	46
4.4.3	Evaluation . . . . .	47
4.5	CSMA-adapté . . . . .	47
4.5.1	Analyse des résultats . . . . .	48
4.5.2	Interprétation des résultats . . . . .	49
4.5.3	Evaluation . . . . .	49
4.6	Transmission sans acquittement . . . . .	50
4.6.1	Analyse des résultats . . . . .	51
4.6.2	Interprétation des résultats . . . . .	52
4.6.3	Evaluation . . . . .	52
4.7	Comparaison selon l'intervalle de régénération . . . . .	53
4.8	Evaluation finale . . . . .	54
4.9	Conclusion . . . . .	56
	<b>Conclusion Générale</b>	<b>57</b>
	<b>Bibliographie</b>	<b>viii</b>

## TABLE DES FIGURES

1.1	Topologie en bus. . . . .	4
1.2	Topologie en étoile. . . . .	5
1.3	Topologie en anneau. . . . .	5
1.4	Architecture client/serveur [2]. . . . .	6
1.5	Architecture poste à poste [2]. . . . .	7
1.6	Architecture générale d'un capteur [4]. . . . .	10
1.7	Réseaux de capteurs [5]. . . . .	11
1.8	Modèle en couche du réseau de capteur [5]. . . . .	11
1.9	Topologies des réseaux de capteurs [7]. . . . .	14
2.1	Liste des principales normes IEEE 802. [12] . . . . .	21
2.2	Principe du CSMA/CD. . . . .	22
2.3	Topologies supportées dans IEEE 802.15.4 [12]. . . . .	24
2.4	Accès au médium en mode CSMA/CA [17]. . . . .	28
2.5	la Procédure CSMA/CA . . . . .	29
2.6	CSMA/CA Slotted et Unslotted dans IEEE 802.15.4 [12]. . . . .	32
3.1	Exemple de messages retardataires. . . . .	35
3.2	Architecture de contiki. . . . .	36
3.3	Le fonctionnement de contikiMAC lors d'un unicast [18]. . . . .	37
3.4	Le fonctionnement de contikiMAC lors d'un broadcast [18]. . . . .	38
3.5	Etablissement de communication entre le mobile et les ancres. . . . .	41
3.6	Le fonctionnement de csma-adapté. . . . .	42
4.1	Interface de simulateur cooja. . . . .	44
4.2	Messages reçus à temps et en retard avec csma-contiki. . . . .	46

4.3	Consommation d'énergie avec csma-contiki. . . . .	46
4.4	Messages à temps et en retard par csma-adapté. . . . .	48
4.5	Consommation d'énergie avec csma-adapté. . . . .	49
4.6	Comparaison des messages à temps et en retard entre csma-contiki /csma-adapté.	50
4.7	Messages à temps et en retard sans ack. . . . .	51
4.8	Consommation d'énergie avec transmission non acquittée. . . . .	52
4.9	Comparaison des messages à temps entre csma-contiki /csma-adapté /sans ack. .	54
4.10	Comparaison des messages retardataires entre csma-contiki /csma-adapté /sans ack. . . . .	54
4.11	Comparaison de la consommation d'énergie entre csma-contiki /csma-adapté /sans ack. . . . .	55

## LISTE DES TABLEAUX

1.1	Les trois générations des nœuds de capteurs. . . . .	9
1.2	comparaison entre les système d'exploitation. . . . .	17
4.1	Résultats de test sur csma-contiki. . . . .	45
4.2	Résultats de test sur csma-adapté. . . . .	48
4.3	Résultats de test sur transmission non acquittée. . . . .	51
4.4	Résultats de test avec variation d'intervalle. . . . .	53

## LISTE DES ABRÉVIATIONS

<b>ACK</b>	ACKnowledgment (Acquittement)
<b>CA</b>	Collision Avoidance
<b>CCA</b>	Clear Channel Accessment
<b>CD</b>	Collision Detection
<b>CSMA</b>	Carrier Sense Multiple Access
<b>CW</b>	Contention Window
<b>DIFS</b>	Distributed Inter Frame Spacingt
<b>FFD</b>	Full Fonction Device
<b>FIFO</b>	First In First Out
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>IETF</b>	Internet Engineering Task Force
<b>IP</b>	Internet Protocol
<b>ISO</b>	International Organization for Standardization
<b>LAN</b>	Local Area Network
<b>LLC</b>	Logical Link Control
<b>LQI</b>	Link Quality Indication
<b>MAN</b>	Metropolitan Area Network
<b>MNS</b>	Multimod AI system for NeTworks of In-situ wireless Sensors
<b>MAC</b>	Medium Access Control
<b>PC</b>	Personnel Computer
<b>OSI</b>	Open System Interconnection
<b>PAN</b>	Personal Area Networks
<b>PDA</b>	Personal Digital Assistant
<b>QoS</b>	Quality of Service
<b>RAM</b>	Random Access Memory

<b>RCSF</b>	<b>R</b> éseau de <b>C</b> apteurs <b>S</b> ans <b>F</b> il
<b>RFD</b>	<b>R</b> educed <b>F</b> unction <b>D</b> evice
<b>ROM</b>	<b>R</b> ead <b>O</b> nly <b>M</b> emory
<b>SIFS</b>	<b>S</b> hort <b>I</b> nter <b>F</b> rame <b>S</b> pacingt
<b>TCP</b>	<b>T</b> ransport <b>C</b> ontrol <b>P</b> rotocol
<b>WAN</b>	<b>W</b> ide <b>A</b> rea <b>N</b> etwork
<b>WSN</b>	<b>w</b> ireless <b>s</b> ensors <b>N</b> etwork

# INTRODUCTION GÉNÉRALE

Le domaine des réseaux informatiques a connu plusieurs nouvelles technologies, parmi ces technologies, on trouve les réseaux de capteurs sans fil (RCSF).

Un réseau de capteur sans fil est un ensemble de micro-capteurs appelés aussi nœuds qui sont dispersés aléatoirement à travers une zone géographique, appelée champ de captage, ces nœuds sont capables de récolter et de transmettre des données d'une manière autonome à partir d'un environnement donné et de transformer ces données en grandeurs numériques dans le but d'établir une communication avec les autres capteurs du réseau et d'acheminer la somme des données collectées vers une station de base.

Les applications des réseaux de capteurs sans fil sont nombreuses. Elles comprennent différents domaines : agricole, militaire...etc. Parmi les problèmes rencontrés dans ce domaine, on trouve la limitation de la durée de vie des capteurs car la durée de vie d'un capteur dépend de sa source d'alimentation qui est la batterie, donc il est important de bien exploiter cette source.

Et aussi parmi ces problèmes, celui des collisions engendrées sur le canal de transmission, ces collisions sont provoquées lorsqu' un ensemble de capteurs sollicitent d'accéder au support dans le but de transmettre leurs paquets de données ce qui provoque des messages retardataires qui sont indésirables par le fait qu'ils ne sont pas reçus à temps, ce qui rend l'énergie consommée pour les transmettre gaspillées.

Le but de notre projet est de proposer des solutions pour remédier ces problèmes.

Dans notre travail nous utilisons le système d'exploitation Contiki dédié aux réseaux de capteurs. Ce système propose une implémentation du protocole d'accès au support CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance) qui n'est pas adapté pour les applications en temps réel. C'est ce que nous allons améliorer, en proposant de nouvelles modifications à ce protocole permettant de pallier ce problème, par un ensemble de tests, en simulations. Pour cela notre mémoire est organisé comme suit :

– **Premier chapitre** : donne un aperçu sur les réseaux informatiques en général et sur les

réseaux de capteurs sans fil en particulier, présente leurs caractéristiques, leurs architectures, leurs systèmes d'exploitation et leurs différents domaines d'applications

- **Deuxième chapitre** : décrit la couche MAC (Medium Access Control) en se basant sur le standard IEEE 802.15, et donne une description détaillée du protocole d'accès au support CSMA/CA.
- **Troisième chapitre** : présente la problématique et l'objectif de notre travail, puis donne plus de détails sur le système d'exploitation Contiki, il présente notre solution proposée csma-adapté et explique son fonctionnement et les modifications apportées.
- **Quatrième chapitre** : est consacré pour effectuer les simulations et tester les solutions que nous avons proposé qui ont pour but d'apporter une amélioration au protocole CSMA/CA implémenté dans Contiki.

On termine le projet par une conclusion générale et perspectives .

# CHAPITRE 1

## INTRODUCTION AUX RÉSEAUX DE CAPTEURS

### 1.1 Introduction

Depuis les premières inventions des systèmes informatiques jusqu'à nos jours, on constate que chaque époque marque une évolution technologique. Actuellement, la technologie des réseaux et les domaines qu'elle inclut prennent de plus en plus d'ampleur dans les systèmes informatiques.

Parmi les technologies, en vogue, répondant à ces domaines, on cite entre autres les réseaux de capteurs sans fil. Un réseau de capteurs sans fil peut être défini comme un ensemble de composants miniatures (capteurs) capable de capter des grandeurs physiques à partir d'un environnement donné et de transformer ces données en grandeurs numériques dans le but d'établir une communication avec les autres capteurs du réseau et d'acheminer la somme des données collectées vers une station de base.

Ce premier chapitre va s'attarder sur les réseaux en générale et les réseaux de Capteurs Sans Fil (RCSFs) ou Wireless Sensor Network (WSN) en partant de leur cellule élémentaire, le capteur sans fil.

### 1.2 Réseaux informatiques

#### 1.2.1 Définition d'un réseau

Un réseau est un ensemble d'objets (imprimantes, ordinateurs, etc.) interconnectés les uns aux autres. Il permet la circulation des informations entre chacun de ces objets selon des règles bien définies. Ceci, dans le but de permettre aux utilisateurs de partager des ressources et des données [1].

## 1.2.2 Intérêt d'un réseau

Un réseau informatique peut être utilisé pour plusieurs objectifs [1] :

- Le partage des fichiers, des applications et des périphériques (imprimantes, disques durs, etc.).
- La communication entre personnes, grâce au courrier électronique, vidéo conférence, etc.
- La communication entre processus, par exemple dans le cas des machines industrielles.

## 1.2.3 Topologie d'un réseau

Pour assurer le bon fonctionnement d'un réseau, c'est-à-dire la circulation correcte de l'information entre les différents dispositifs matériels de ce réseau, nous parlerons de topologie logique et physique [2].

### 1.2.3.1 Topologie logique

Elle représente la façon dont les données transitent dans les lignes de communication. Les topologies logiques les plus courantes sont : Ethernet, Token Ring et FDDI (le protocole qui assure l'interconnexion des réseaux locaux et des serveurs).

### 1.2.3.2 Topologie physique

on distingue trois topologies principales :[2]

- **Topologie en bus** : Dans le cas d'un réseau utilisant une topologie en bus, la connexion est constituée d'un seul fil couvrant l'étendue du réseau. Chaque ordinateur ou périphérique est raccordé à cette ligne de communication.

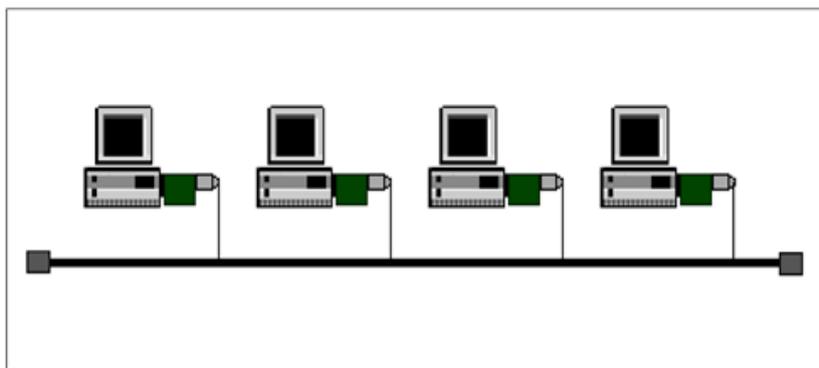


FIGURE 1.1 – Topologie en bus.

- **Topologie en étoile** : C'est la topologie la plus utilisée actuellement, notamment par les réseaux Ethernet. Chaque ordinateur ou périphérique est relié à un point central appelé

concentrateur réseau. Ce dernier, peut être un Switch, un hub, ou un routeur pour la liaison inter-réseaux.

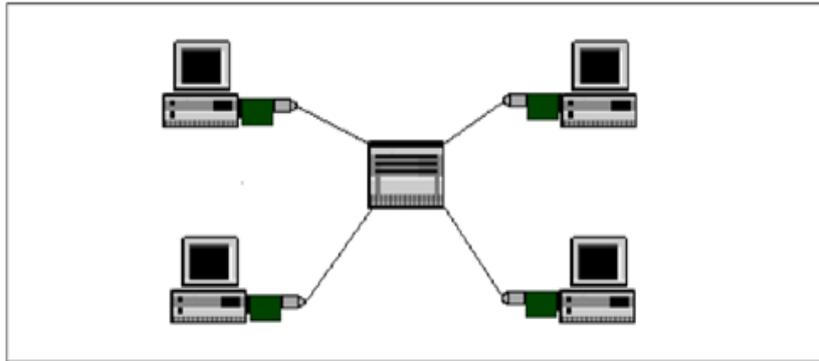


FIGURE 1.2 – Topologie en étoile.

- **Topologie en anneau** : Elle se caractérise par une connexion circulaire de la ligne de communication. Le transfert de données entre deux ordinateurs sur le réseau peut donc se faire suivant deux directions.

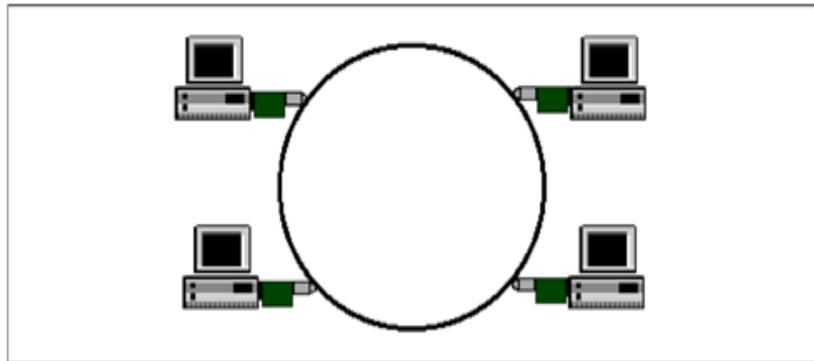


FIGURE 1.3 – Topologie en anneau.

#### 1.2.4 Classification des réseaux

On différencie un réseau par rapport à un autre selon deux critères importants l'étendue du réseau et la capacité et la vitesse de transmission des données. On distingue alors trois types de réseaux [2] :

- **Réseaux LAN** : (Local Area Network) Ces réseaux s'étendent sur une distance de quelques mètres à quelques kilomètres. De tels réseaux offrent, en général, une étendue géographique comprise entre 1 mètre et 1 Kilomètre.

- **Réseaux WAN :** (Métropolitain Area Network)MAN Ces réseaux s'étendent sur une zone géographique de la taille d'une ville. Dans ce type de réseaux, l'étendue géographique est comprise entre 1 kilomètre et 100 kilomètres.
- **Réseaux WAN :** (Wide Area Network)WAN Ces réseaux sont généralement constitués de plusieurs sous-réseaux hétérogènes et s'étendent sur une région ou un pays entier. L'étendue géographique de ce type de réseaux va jusqu'à 100 kilomètres et plus.

### 1.2.5 Architecture des réseaux

Il existe deux architectures principales : l'architecture client/serveur et l'architecture poste à poste.[2]

- **Architecture client/serveur :** L'architecture client/serveur désigne un mode de communication entre plusieurs ordinateurs d'un réseau. Ce réseau est composé d'un ou plusieurs clients et d'un serveur, chaque client peut envoyer des requêtes au serveur qui peut être, soit un serveur d'applications, de fichiers, soit de messagerie électronique.

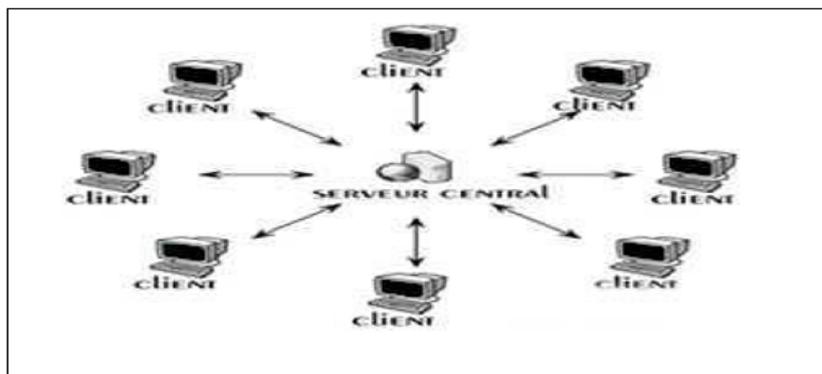


FIGURE 1.4 – Architecture client/serveur [2].

- **Architecture poste à poste :** Dans une architecture poste à poste, contrairement à une architecture client/serveur, il n'y a pas de serveur dédié, c'est à dire que chaque ordinateur est à la fois serveur et client. Cela signifie que chaque ordinateur du réseau est libre de partager ses ressources.

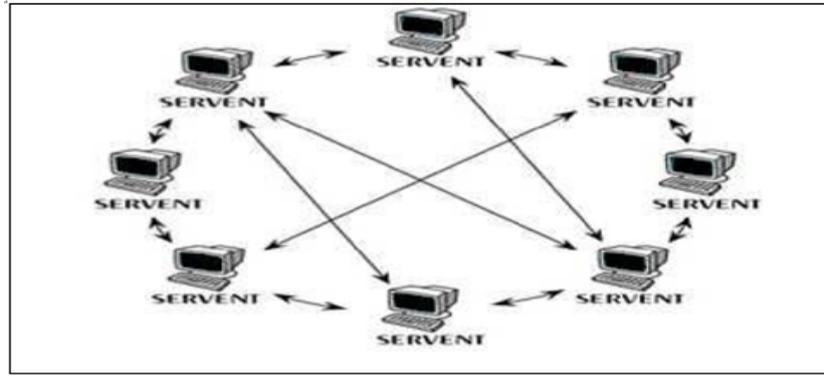


FIGURE 1.5 – Architecture poste à poste [2].

### 1.2.6 Modèles de communication OSI

Le modèle OSI (Open System Interconnexion), signifie interconnexion des systèmes ouverts. Il définit la manière dont les ordinateurs et les périphériques doivent procéder pour communiquer.

Le modèle OSI se décompose en sept parties appelées « couches ». Chaque couche est responsable de l'un des aspects de la communication. Une couche de niveau  $N$  communique avec les couches de niveau  $N+1$  et  $N-1$  par le biais d'une interface. Chaque couche inférieure fournit des services pour la couche supérieure, les couches  $N$  de deux systèmes communiquent à l'aide des protocoles de communication communs.[1]

#### Sept couches du modèle OSI

Les sept couches du modèle OSI se présentent comme suit :[1]

- **Couche physique** : cette couche se charge de l'adaptation du signal au support de transmission, elle gère aussi le type de transmission du signal (mode synchrone ou asynchrone).
- **Couche liaison de données** : cette couche définit les règles d'émission et de réception de données, ainsi que la mise en œuvre de la détection et de la correction d'erreurs. Elle gère également le contrôle de flux.
- **Couche réseaux** : elle gère l'acheminement des données en assurant le routage des paquets de données et la traduction des adresses logiques en adresses physiques.
- **Couche transport** : elle se charge du transport de données et des services qui n'ont pas été pris en compte par les couches inférieures tels que la gestion des erreurs, le routage des paquets de données, etc.
- **Couche session** : cette couche permet l'ouverture et la fermeture d'une session de travail entre deux systèmes distants, elle assure la synchronisation du dialogue et définit également le mode de transmission (simplex, half-duplex, full-duplex, etc.).

- **Couche présentation** : cette couche assure la mise en forme des données pour qu'elles soient accessibles à l'utilisateur. Elle effectue les fonctions de codage, compression, cryptage, décryptage, etc.
- **Couche application** : elle se charge du transfert des fichiers, de la messagerie électronique et prend en charge la lecture des pages web.

## 1.2.7 Type des réseaux

On trouve deux types qui sont les suivants :[2]

### 1.2.7.1 Réseaux filaires

C'est un réseau dont la communication entre les terminaux se fait avec une liaison filaire grâce au câble (câble coaxiale, la fibre optique, paires torsadées....).

### 1.2.7.2 Réseaux sans fil

Un réseau sans fil (en anglais Wireless network) est, comme son nom l'indique, un réseau dans lequel au moins deux terminaux (ordinateur portable, pDA, etc.) peuvent communiquer sans liaison filaire.

## 1.3 Réseaux de capteur sans fil

### 1.3.1 Historique des réseaux de capteurs sans fil

Les récents progrès des nouvelles techniques ont provoqué une énorme importance dans le domaine des réseaux sans fil. La technologie des réseaux de capteurs sans fil est devenue une des merveilleuses technologies dans le 21ème siècle , les réseaux de capteurs ont montré leur impact sur notre vie quotidienne.

Le tableau suivant illustre l'évolution des réseaux de capteurs [3].

Génération	Période	Taille	Poids	Batterie
1 ère	Les années 80 et 90	Grande boîte à chaussures	Kilogrammes	Grosse
2 ème	Entre 2000 et 2003	Boîte de cartes	Grammes	AA
3 ème	2010	Particule de poussière	Négligeable	Solaire

TABLE 1.1 – Les trois générations des nœuds de capteurs.

Grâce aux progrès des techniques sans fil, les réseaux de capteurs seront aussi communs dans notre vie quotidienne comme les ordinateurs et l'internet.

## 1.3.2 présentation des réseaux de capteurs

### 1.3.2.1 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.[4]

### 1.3.2.2 Architecture d'un nœud (capteur)

Tous les capteurs respectent globalement la même architecture basée sur un noyau central autour duquel s'articulent les différentes interfaces d'entrée sortie.

On peut voir sur la figure suivante les différents composants qui constituent un capteur, pour être plus précis chaque groupe de composants possède son propre rôle de communication et d'alimentation.

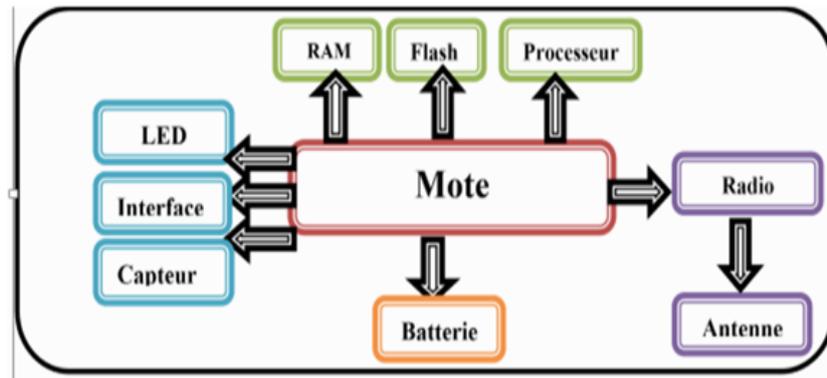


FIGURE 1.6 – Architecture générale d'un capteur [4].

- **Mote, processeur, RAM et Flash** : On appelle généralement mote la carte physique utilisant le système d'exploitation pour fonctionner. Celle-ci a pour cœur le bloc constitué du processeur et des mémoires RAM et Flash. Cet ensemble est à la base du calcul binaire et du stockage, temporaire pour les données et définitif pour le système d'exploitation.
- **Radio et antenne** : Un capteur est conçu pour mettre en place des réseaux sans fils, les équipements étudiés sont donc généralement équipés d'une radio ainsi que d'une antenne afin de se connecter à la couche physique que constitue les émissions hertziennes.
- **LED, interface, capteur** : Prévus pour mettre en place des réseaux de capteurs, on retrouve donc des équipements bardés de différents types de détecteurs et autres entrées.
- **Batterie** : Comme tout dispositif embarqué, il dispose d'une alimentation autonome telle qu'une batterie, et parfois d'un panneau solaire pour permettre de recharger cette batterie, ce qui lui permet d'être disposé dans un endroit parfois inaccessible.

Cependant quelques différences existent suivant les fabricants. Chacun d'eux développe son type de capteurs, ces types peuvent être Mica, Mica2, Telos ou Telosb par exemple.

Avec des contraintes hardware aussi strictes dues à la miniaturisation des capteurs, la partie software doit être la plus adaptée possible, d'où un lien très fort entre ces deux parties.

### 1.3.3 Définition d'un RCSF ou WSN (Wireless Sensor Network)

Les réseaux de capteurs utilisent un grand nombre de dispositifs très petits, nommés (nœuds capteurs), pour former un réseau sans infrastructure établie. Dans ces réseaux, chaque nœud est capable de détecter son environnement et de traiter l'information au niveau local ou de l'envoyer à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil, 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).

Le réseau possède en général un nœud particulier, la base (ou sink), connectée avec les autres nœuds et relié à une alimentation électrique ce qui est illustré par la figure suivante.

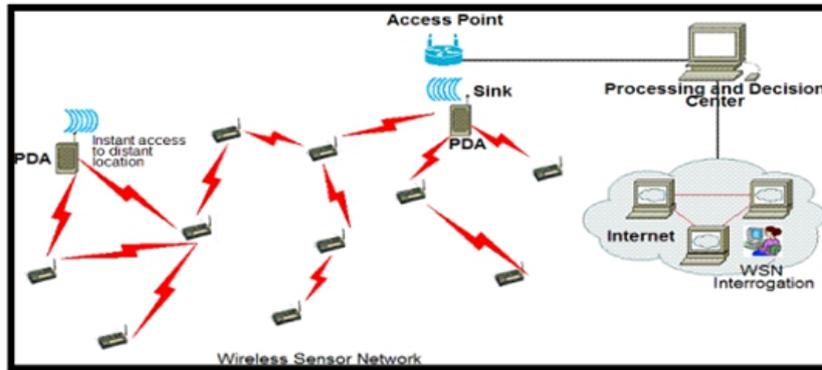


FIGURE 1.7 – Réseaux de capteurs [5].

### 1.3.4 Architecture de communication dans les réseaux de capteurs

Le modèle de communication comprend cinq couches qui ont les mêmes fonctions que celles du modèle OSI ainsi que trois couches pour la gestion d'énergie, la gestion de la mobilité et la gestion des tâches ce que illustre la figure suivante.[5]

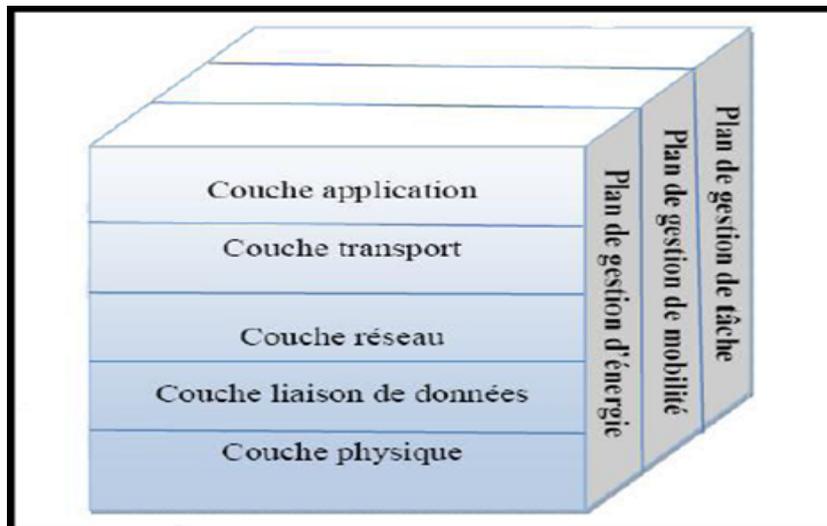


FIGURE 1.8 – Modèle en couche du réseau de capteur [5].

- **Couche physique** : Matériels pour envoyer et recevoir les données.
- **Couche liaison de données** : Gestion des liaisons entre les nœuds et les stations de base, contrôle d'erreurs.

- **Couche réseau** : Routage et transmission des données.
- **Couche transport** : Transport des données, contrôle de flux.
- **Couche application** : la couche application constitue l'ensemble des applications implémentées sur un réseau de capteurs, ces applications doivent fournir des mécanismes permettant à l'utilisateur d'interagir avec le réseau de capteurs à travers différentes interfaces.

Le modèle de communication dans les réseaux de capteurs comprend 03 plans qui sont les suivants

- **Plan de gestion d'énergie** : Contrôle l'utilisation de la batterie. Par exemple, après la réception d'un message, le capteur éteint son récepteur afin d'éviter la duplication des messages déjà reçus. En outre, si le niveau d'énergie devient bas, le nœud diffuse à ses voisins une alerte les informant qu'il ne peut pas participer au routage. L'énergie restante est réservée au captage.
- **Plan de gestion de mobilité** : Puisque les nœuds peuvent être mobiles, un système de gestion de mobilité doit exister. Un tel système doit être capable d'enregistrer les mouvements du nœud à fin de l'aider à se localiser.
- **Plan de gestion de tâche** : Lors d'une opération de captage dans une région donnée, les nœuds composant le réseau ne doivent pas obligatoirement travailler avec le même rythme, cela dépend essentiellement de la nature du capteur, son niveau d'énergie et la région dans laquelle il a été déployé. Pour cela, le niveau de gestion des tâches assure l'équilibrage et la distribution des tâches sur les différents nœuds du réseau, à fin d'assurer un travail coopératif et efficace en matière de consommation d'énergie, et par conséquent, prolonger la durée de vie du réseau.

### 1.3.5 Caractéristiques d'un réseau de capteur sans fils

Parmi les caractéristiques les plus importantes d'un réseau de capteurs, nous citons :[6]

- **Durée de vie limitée** : Les nœuds capteurs sont très limités par la contrainte d'énergie, ils fonctionnent habituellement sans surveillance dans des régions géographiques éloignées. Par conséquent recharger ou remplacer leurs batteries devient quasiment impossible.
- **Ressources limitées** : Habituellement les nœuds capteurs ont une taille très petite, ce facteur de forme limite la quantité de ressources qui peuvent être mises dans ces nœuds. En conséquence, la capacité de traitement et de mémoire est très limitée.
- **Topologie dynamique** : La topologie des réseaux de capteurs change d'une manière fréquente et rapide car les nœuds capteurs peuvent être déployés dans des environnements hostiles (par exemple un champ de bataille), la défaillance d'un nœud capteur peut donc être très probable. De plus, les nœuds capteurs et les nœuds finaux où ils doivent envoyer

l'information capturée peuvent être mobiles.

- **Agrégation des données** : Dans les réseaux de capteurs, les données produites par les nœuds capteurs sont très reliées, ce qui implique l'existence de redondances de données. Une approche répandue consiste à agréger les données au niveau des nœuds intermédiaires afin de réduire la consommation d'énergie lors de la transmission de ces données.
- **Scalabilité** : Les réseaux de capteurs engendrent un très grand nombre de capteurs, ils peuvent atteindre des milliers voire des millions de capteurs. Le défi à relever par les RCFSs est d'être capable de maintenir leurs performances avec ce grand nombre de capteurs.
- **Bande passante limitée** : En raison de la puissance limitée, les nœuds capteurs ne peuvent pas supporter des débits élevés.
- **Sécurité physique limitée** : Cela se justifie par les contraintes et limitations physiques qui minimisent le contrôle des données transmises.

### 1.3.6 Topologies des réseaux de capteurs

Un réseau de capteurs sans fil est composé d'un ensemble de nœuds capteurs et des passerelles qui s'occupent de collecter les données des capteurs et de les transmettre à l'utilisateur via l'internet ou le satellite, il existe plusieurs topologies pour les réseaux de capteurs.[7]

- **Topologie en étoile** : La 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.

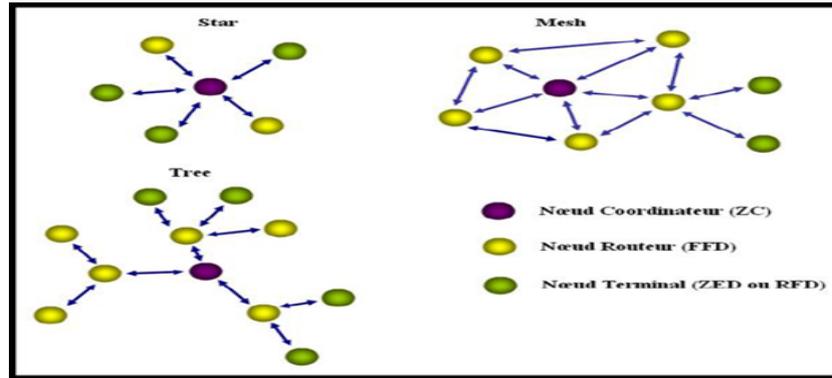


FIGURE 1.9 – Topologies des réseaux de capteurs [7].

### 1.3.7 Systèmes d'exploitation pour les réseaux de capteurs

Les composants à l'intérieur d'un capteur sont de plus en plus diversifiées, ce qui rend leur gestion complexe et difficile. Les plateformes hardwares existantes pour les capteurs sans fil utilisent des modules de communications sans fil différents et des dispositifs de captage distincts, la seule caractéristique qui est unifiée est l'architecture du processeur avec les données en RAM et le code de programme en ROM, d'où le besoin d'une couche logiciel qui interfacera le matériel et les programmes, cette couche s'appelle traditionnellement système d'exploitation et son premier rôle est d'abstraire les applications et donc les programmeurs des différents détails bas niveau liés au matériel.[8]

#### 1.3.7.1 Définition d'un système d'exploitation

Un système d'exploitation peut être vu comme une machine virtuelle au-dessus du matériel plus facile d'emploi et plus conviviale. Vu d'en dessous le système d'exploitation est un gestionnaire de ressources, il prend à sa charge la gestion de ressources de plus en plus complexes et diversifiées.[9]

#### 1.3.7.2 Caractéristiques des systèmes d'exploitation pour RSCF

Les importantes caractéristiques d'un système d'exploitation pour capteurs sans fil sont les suivantes :[9]

- **Temps réel** : une grande partie des applications pour les réseaux de capteurs sont des applications dites temps réel (localisation, détection d'incendie, etc.), d'où la nécessité d'utiliser des systèmes d'exploitations à temps réel car ces derniers sont plus prévisibles par rapport aux systèmes classiques ce qui nous assure qu'une tâche donnée terminera son exécution dans les délais fixés.

- **L’empreinte mémoire** : l’empreinte mémoire d’un système est la quantité de la mémoire occupée par ce dernier. A cause des ressources limitées sur un capteur les systèmes destinés aux réseaux de capteurs doivent avoir l’empreinte mémoire la plus petite possible.
- **Le chargement et le déchargement dynamique** : les capteurs sont souvent déployés à de grandes échelles dans des environnements très durs d’accès. Il est donc primordial d’implémenter au niveau du système d’exploitation des mécanismes qui permettront le chargement et le déchargement individuel des applications et des services sans avoir recours à envoyer une image binaire complète du système et de l’application qui est souvent beaucoup plus grande en taille que l’application et donc coûteuse en énergie pour la transmission.

### 1.3.7.3 Quelques systèmes d’exploitation existants

Parmi les systèmes d’exploitation existants, on présente les plus utilisés :[8]

- **TinyOS** : TinyOS est un système d’exploitation open source conçu pour les capteurs sans fils et développé par l’Université de Berkeley. Il est basé sur une architecture à base de modules : pilotes pour les capteurs, les protocoles réseau et les services distribués.

Les composants sont programmés en NesC, un langage de programmation dérivé du C adapté aux faibles ressources physiques des capteurs. Un certain nombre de plateformes sont directement programmables comme par exemple les tmote ou les MicaZ (ces deux modèles sont compatibles avec ZigBee). TOSSIM est un simulateur de capteurs pour les programmes TinyOS, tout programme en NesC peut être compilé de manière à être exécuté dans TOSSIM, ce qui permet de simuler le comportement d’un ou plusieurs capteurs ainsi de les programmer.[8]

- **MANTIS** : MANTIS (Multimodal system for Network of In-situ Wireless Sensors) est un système d’exploitation multithread préemptif dédié aux RCSFs. Il est développé à l’Université du Colorado avec le langage de programmation C.

Le MANTIS fournit un environnement commode pour créer des applications pour les RCSFs, grâce à L’API du système. Grâce à elle nous pouvons facilement créer des applications qui s’exécuteront comme des threads, une application peut créer un autre thread à l’aide de la primitive `thread-new ()`, MANTIS initialise correctement d’autres thread du système, tels que les threads de la pile de réseau par exemple.

Les entités COMM et DEV offrent une interface aux threads pour communiquer avec les périphériques. Les entités COMM est en charge des communications synchrones et DEV des Communications asynchrones.[10]

- **Contiki** : Contiki est un système d’exploitation libre, léger, multitâche, hautement portable, développé pour des appareils possédant peu de mémoire, allant de l’ordinateur 8

bits aux systèmes embarqués sur des microcontrôleurs, dont les réseaux de capteurs sans fil.

Contiki a présenté l'idée d'utiliser la communication IP dans des réseaux de capteurs basse consommation. Contiki contient deux piles de communication : uIP et Rime : UIP est une petite pile de TCP/IP RFC-CONFORME qui permet à Contiki de communiquer sur Internet. Rime est une pile de communication légère conçue pour des radios basse puissance. Il fournit une vaste gamme de communications primitives.

Contiki fonctionne sur une grande variété de plate-forme comme le MSP430, l'AVR, de vieux ordinateurs familiaux... L'empreinte de code est de l'ordre du kilo-octet et l'utilisation de la mémoire peut être configurée jusqu'à un minimum de quelques dizaines d'octets. Contiki est écrit en langage C et est librement disponible étant donné que la source est ouverte conformément à une licence BSD.[10]

### 1.3.7.4 Comparaison entre ces systèmes d'exploitation

Le tableau suivant illustre une comparaison entre ces trois systèmes d'exploitation pour réseaux de capteurs [10].

	TinyOS	Contiki	Mantis
Publication (année)	ASPLOS (200)	EmNets (2004)	MONET (2005)
Statique/Dynamique	Statique	Dynamique	Dynamique
Evènementiel/Thread	Event& Thread (TinyThread,TOSThreads)	Event& Threads (Protothreads)	Thread & Event (TinyMOS)
Monolithique/Modulaire	Monolithique	Modulaire	Modulaire
Réseau	Message Actif	uIP,uIPv6,Rime	"comm"
Langage de programmation	nesC	C	C
Système de fichier	Un seul niveau	Coffee	Non
Reconfiguration	Oui	Oui	Non
Débugage distant	Oui	Non	Oui

TABLE 1.2 – comparaison entre les système d'exploitation.

En comparant les caractéristiques importantes de TinyOS, Contiki et Mantis, il en ressort que Contiki et Mantis, avec leurs systèmes dynamiques et modulaires sont, contrairement à TinyOS basé sur un système statique et monolithique, plus flexibles et donc plus adaptés pour un changement d'environnement dynamique ou dans le cas d'une reprogrammation au travers du réseau.

Le système d'exploitation de TinyOS et Contiki sont écrits en C, contrairement à TinyOS écrit en NesC ce qui les rend très portables. L'implémentation du mécanisme de multithreads est également différente puisque Mantis la gère nativement, TinyOS uniquement grâce à sa librairie TinyThreads, et contiki obtient ce mode d'exécution soit par librairie, soit par protothreads Le protothread de contiki réduit l'utilisation mémoire, au détriment de certaines fonctionnalités comme le maintien des variables locales.

La pile réseau de TinyOS est la plus légère, elle est basée sur le principe de messages pondérés. Contiki quant à lui contient 2 couches de communication, Rime et uIP qui lui permettent de communiquer avec les protocoles de l'internet, y compris en IPv6. Contiki implémente uIPv6, la plus petite couche IPv6 au monde, utilisable dans le domaine des capteurs sans fils. Et pour la configuration, elle est possible dans les deux systèmes TinyOS et Contiki non pas dans Mantis.

### 1.3.8 Domaines d'application des réseaux de capteurs

Les applications des réseaux de capteurs sont nombreuses et variées, elles peuvent être placées en quatre grandes familles d'applications : militaire, environnementale, santé et les applications commerciales. Pour chaque application un cahier de charge est défini, il comprend le nombre et le type de capteurs appropriés, la topologie adéquate, etc. Cette section propose de découvrir une partie de ces applications.[13]

#### 1.3.8.1 Domaine militaire

Comme pour de nombreuses autres technologies, le domaine militaire a été le moteur initial pour le développement des réseaux de capteurs. Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui font de ce type de réseaux un outil appréciable dans un tel domaine. Actuellement, les RCSFs peuvent être une partie intégrante dans le commandement, le contrôle, la communication, la surveillance, la reconnaissance, etc.

#### 1.3.8.2 Domaine commerciale

Parmi les domaines dans lesquels les réseaux de capteurs ont aussi prouvé leur utilité, on trouve le domaine commercial. Dans ce secteur on peut énumérer plusieurs applications comme : la surveillance de l'état du matériel, le contrôle et l'automatisation des processus d'usinage, etc.

#### 1.3.8.3 Domaine environnementale

Dans ce domaine, les capteurs peuvent être exploités pour détecter les catastrophes naturelles (feux de forêts, tremblements de terre, etc.), détecter des émanations de produits toxiques (gaz, produits chimiques, pétrole, etc.) dans des sites industriels tels que les centrales nucléaires ou pétrolières.

#### 1.3.8.4 Domaine médical

Le domaine médical peut lui aussi intégrer des applications pertinentes. Comme par exemple : l'aide à la médication et le suivi des patients à distance (rythme cardiaque, pres-

sion du sang, etc), l'identification des allergies et des médicaments administrés aux patients, la localisation des docteurs et des patients dans l'hôpital, etc.

## 1.4 Conclusion

Les réseaux de capteurs sans fil présentent un intérêt considérable et une nouvelle étape dans l'évolution des technologies de l'information et de la communication. Cette nouvelle technologie suscite un intérêt croissant vu la diversité de ces applications : santé, environnement. Dans ce chapitre nous avons abordé l'architecture matérielle d'un capteur, et nous avons présenté les réseaux de capteurs sans fil en exposant leurs architectures, leurs domaines d'application ainsi que leurs systèmes d'exploitation.

Nous consacrons le prochain chapitre pour parler sur le standard IEEE 802.15 qui s'articule sur la couche MAC du modèle OSI où décrit le protocole d'accès au support CSMA/CA utilisé par les réseaux locaux sans fil et c'est ce qui nous intéresse pour ce travail.

## CHAPITRE 2

# LA COUCHE MAC DANS IEEE.802.15.4

### 2.1 Introduction

Le monde des réseaux et de l'internet compte deux organismes de standardisations très influents : L'IEEE et IETF. Ces deux organismes ont tous deux comme objectif de standardiser [12] :

- les couches basses du modèle OSI (physique et MAC) pour l'IEEE.
- Les couches hautes pour l'IETF (réseau, transport, etc).

L'objectif de ce chapitre est de présenter en premier lieu les différentes normes de l'IEEE en mettant le point sur les protocoles d'accès au support, en deuxième lieu nous détaillerons le standard 802.15 le plus utilisé dans les réseaux locaux sans fil. Ensuite, nous présenterons les caractéristiques liées à l'architecture logique de la norme (couche physique et couche MAC).

### 2.2 Normes IEEE 802

#### 2.2.1 Standard IEEE

Les réseaux informatiques doivent permettre à des applications informatiques de coopérer sans avoir à tenir compte de l'hétérogénéité des moyens et procédés de transmission (par exemple : de la topologie, des méthodes d'accès, des caractéristiques des équipements ou des supports, etc.). La normalisation est un ensemble de règles établies qui doivent être suivies par les entités désirant communiquer. En 1980, l'IEEE a créé le comité d'étude 802, chargé de définir des normes pour les réseaux locaux, afin d'assurer la compatibilité entre les équipements provenant de différents constructeurs. Une famille de normes a été élaborée. [12] Toutes les normes du comité IEEE 802 ont été reprises et complétées par l'ISO (Organisation Internatio-

nale de Normalisation) sous la désignation ISO 802, elles correspondent aux couches physique et liaison de données du modèle de référence OSI. L'IEEE a défini les fonctionnalités de la sous-couche LLC (Logical Link Control) dans la norme 802.2 et celles de la sous-couche MAC (couche Physique) dans les normes 802.3, 802.4 et 802.5. L'architecture IEEE 802 est décrite par la figure 2.1

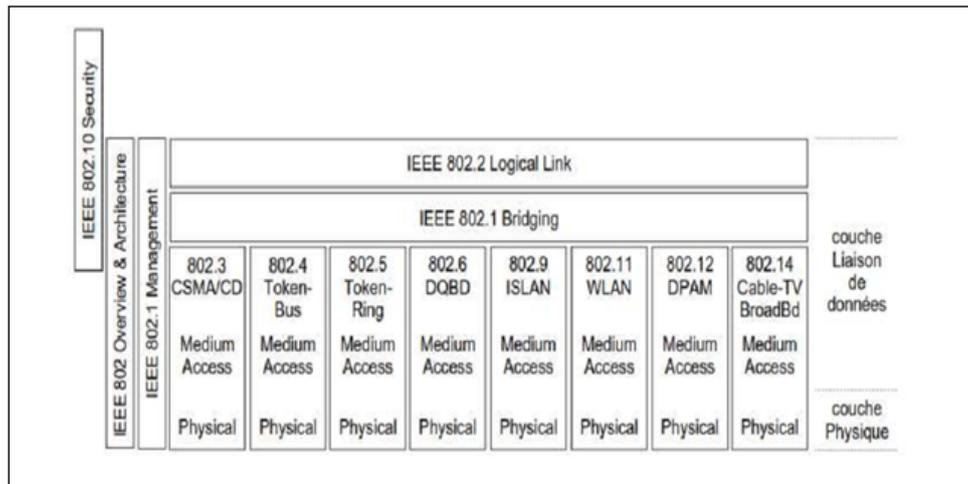


FIGURE 2.1 – Liste des principales normes IEEE 802. [12]

## 2.2.2 Normes IEEE 802.x

Le comité 802 n'a pas donné naissance à un unique standard, mais à une série de standards pour couvrir l'ensemble des besoins. C'est ainsi que plusieurs sous-comités ont été créés chacun traitant d'un sujet particulier dans lequel une norme est élaborée.

### 2.2.2.1 Norme 802.3

La norme 802.3 définit des réseaux locaux dont le support choisi est un bus logique auquel sont connectés tous les éléments actifs et utilisant le protocole d'accès au support CSMA/CD (Carrier Sense Multiple Acces/ Collision Detection), tel que tous les ordinateurs du réseau 'écoutent' le support afin de savoir s'il y a du trafic sur le réseau :

- Un ordinateur détecte que ce câble est libre (il n'y a pas de trafic).
- L'ordinateur peut émettre des données.
- Si les données se trouvent déjà sur le câble, aucun autre ordinateur ne peut transmettre tant que les données n'ont pas atteint leur destination et que le câble n'a pas été libéré.
- Si deux (ou plusieurs) ordinateurs envoient des données sur le câble au même instant, il y aura collision. En pareil cas, les ordinateurs cessent de transmettre pendant une durée

aléatoire, puis ils essaient de reprendre la transmission. Après 16 tentatives d'émission d'un même message, l'émetteur abandonne l'émission, la figure suivante illustre ce principe.

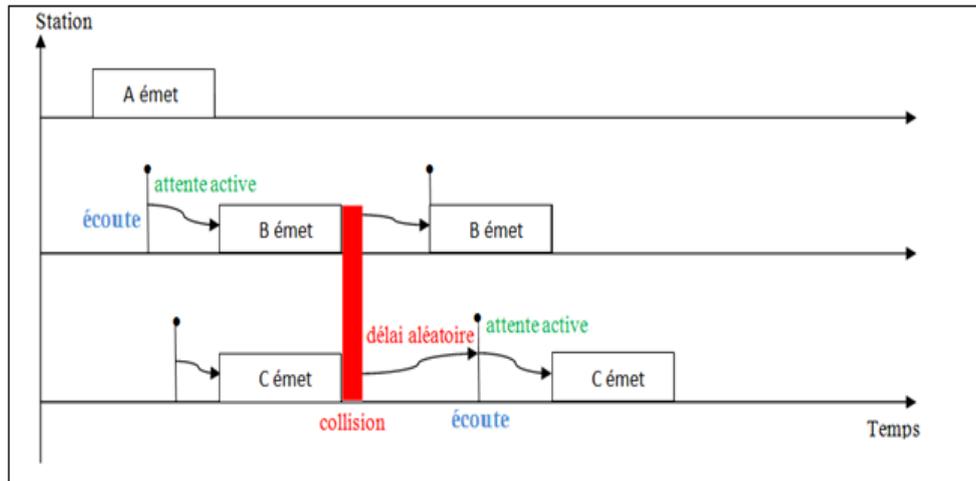


FIGURE 2.2 – Principe du CSMA/CD.

### 2.2.2.2 Norme 802.11

L'IEEE a normalisé une architecture de réseaux locaux sans fil basée sur une technologie radio utilise CSMA/CA (Carrier Sense Multiple Acces/ Collision Avoidance), comme protocole d'accès au support. Cette norme a été nommée 802.11. La norme IEEE 802.11 se décompose en plusieurs normes de transmission, offrant chacune des caractéristiques différentes en terme de fréquence, de débit ou de portée du signal radio.[14]

### 2.2.2.3 Norme 802.12

La norme IEEE 802.12 utilise le mécanisme de demande de priorité pour l'accès aux médias à 100 Mbit/s. Elle est aussi appelée 100VG-Any-LAN. Où :

- VG (Voice Grade) : Priorité à la demande.
- Any-LAN : Accepte tous les réseaux (Trame Ethernet ou Token Ring).

### 2.2.2.4 Norme 802.15

Le groupe IEEE 802.15, intitulé WPAN (Wireless Personal Area Networks), a été mis en place en mars 1999 dans le but de réfléchir aux réseaux d'une portée d'une dizaine de mètres, a pour objectif de réaliser des connexions entre les différents portables d'un même utilisateur ou de plusieurs utilisateurs. Ce réseau peut interconnecter un PC portable (lap top), un téléphone

portable, un PDA ou tout autre terminal de ce type. Trois groupes de services ont été définis A, B et C [12].

- **Groupe A** : utilise la bande du spectre sans licence d'utilisation (2,4 GHz) en visant un faible coût de mise en place et d'utilisation. La taille de la cellule autour du point d'émission est de l'ordre du mètre. La consommation électrique doit être particulièrement faible pour permettre au terminal de tenir plusieurs mois sans recharge électrique. Le mode de transmission choisi est sans connexion. Le réseau doit pouvoir travailler en parallèle d'un réseau 802.11, c'est-à-dire que sur un même emplacement physique il peut y avoir en même temps un réseau de chaque type, les deux pouvant éventuellement fonctionner de façon dégradée.
- **Groupe B** : affiche des performances en augmentation avec un niveau MAC pouvant atteindre un débit de 100 Kbit/s. Le réseau de base doit pouvoir interconnecter au moins seize machines et proposer un algorithme de QoS, ou qualité de service, pour autoriser le fonctionnement de certaines applications, comme la parole téléphonique, qui demande une qualité de service assez stricte. La portée entre l'émetteur et le récepteur atteint une dizaine de mètres, et le temps maximal pour se raccorder au réseau ne doit pas dépasser la seconde. En fin, cette catégorie de réseau doit posséder des passerelles avec les autres catégories de réseaux 802.15.
- **Groupe C** : introduit de nouvelles fonctionnalités importantes pour particuliers ou entreprises, comme la sécurité de la communication, la transmission de la vidéo et la possibilité de roaming, ou itinérance, entre réseaux hertziens. Pour répondre à ces objectifs, des groupements industriels se sont mis en place, comme Bluetooth. Bluetooth regroupe plus de 2 500 sociétés qui ont réalisé une spécification ouverte de connexion sans fil entre équipements personnels. Bluetooth est fondé sur une liaison radio entre deux équipements.
- IEEE 802.15.3, pour les contraintes posées par le groupe B, mais qui a finalement débouché sur une proposition très performante avec l'UWB (Ultra-Wide Band), qui était sur le marché en 2005.
- IEEE 802.15.2, pour les interférences avec les autres réseaux utilisant la bande des 2,4 GHz.
- IEEE 802.15.4, pour les réseaux WPAN de catégorie A, qui a abouti à la proposition ZigBee, d'un réseau à bas débit mais à un coût extrêmement bas.

## 2.3 IEEE 802.15.4

Le standard IEEE 802.15.4 définit deux types de nœuds : FFD (Full Fonction Device) et RFD (Reduced Fonction Device) pouvant jouer le rôle de :

- PAN Coordinateur qui est le nœud initiateur du réseau. Il a comme attribution d'initier, de gérer l'ensemble du réseau et de jouer le rôle de passerelle vers l'extérieur.
- Coordinateur qui est par définition un FFD et qui aura pour tâche de gérer une zone donnée du réseau (cluster) et de router les paquets de ses fils vers le PAN coordinateur.
- Nœuds terminaux qui sont de simples nœuds périphériques (actionneurs, capteurs). On peut définir les FFD et RFD comme suit :
- **FFD (Full-FunctionDevices)** : Peuvent communiquer avec des FFD et des RFD, et peuvent ainsi jouer le rôle de coordinateur de PAN (puits), de routeur (Coordinateur) ou de simple périphérique (capteur, interrupteur).
- **RFD (Reduced-FunctionDevices)** : Sont des nœuds terminaux utilisés dans des contextes simples comme les interrupteurs ou de simples capteurs. Ces nœuds ne peuvent communiquer qu'avec des FFD (coordinateur) et ne peuvent participer au routage dans le réseau. Ils ne sont ainsi que des nœuds terminaux dans le réseau avec des ressources moindres en comparaison avec des FFD.

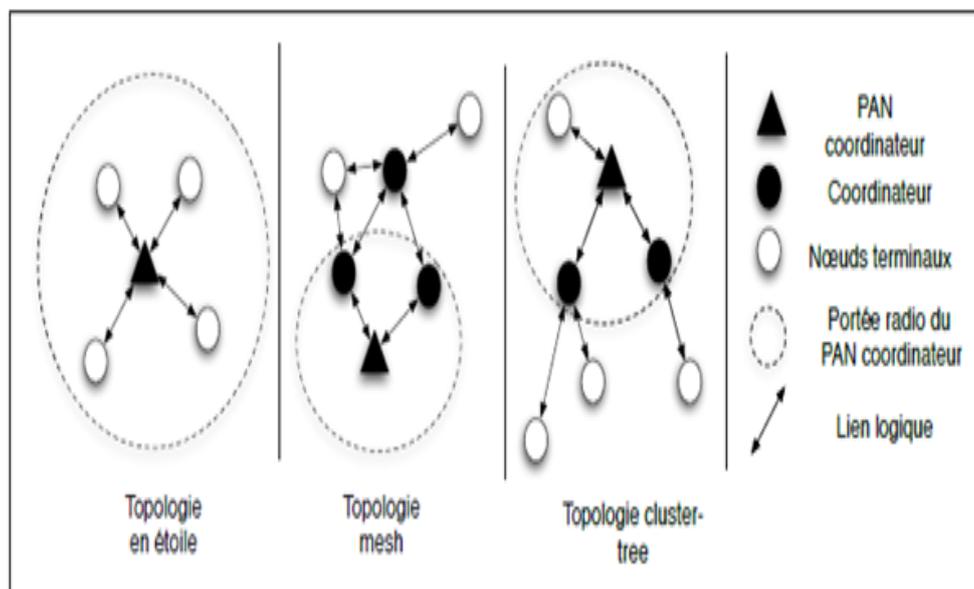


FIGURE 2.3 – Topologies supportées dans IEEE 802.15.4 [12].

Le standard IEEE 802.15.4 supporte plusieurs types de topologies (figure 2.3) : La topologie en étoile, la topologie point à point (Peer-to-Peer) et la topologie en arbre (cluster-tree). Il est possible d'avoir plusieurs réseaux PAN opérant dans la même zone mais fonctionnant indépendamment les uns des autres, chacun utilisant un identifiant de PAN différent.

- **Topologie en étoile** : impose des communications s'établissant directement et uniquement entre le nœud qui gère le réseau défini précédemment (PAN coordinateur) et les nœuds se trouvant dans sa portée radio. Le PAN coordinateur sera le nœud initiateur du réseau et tout le trafic devra passer par ce nœud.
- **Topologie point à point** : chaque nœud pourra communiquer avec n'importe quel autre nœud du réseau grâce à la collaboration des nœuds intermédiaires (Coordinateur) sollicités afin de relayer les paquets jusqu'à la destination. Cette topologie permet de réaliser des réseaux beaucoup plus étendus, mais il faudra pour cela ajouter un protocole de routage.
- **Topologie cluster-tee** : n'est qu'un cas particulier de la topologie Peer-to-Peer. La seule différence avec cette dernière est que le réseau sera hiérarchisé avec un PAN coordinateur, des coordinateurs (têtes de cluster) gérant les nœuds qui leur sont connectés, ainsi que des nœuds terminaux communiquant uniquement avec leur coordinateur.

Le standard IEEE 802.15.4 ne définit pas de méthode pour la création de topologie, il définit uniquement les topologies pouvant être utilisées. La méthode de création de topologie est laissée à la libre interprétation du développeur. Néanmoins la méthode d'association des nœuds au réseau a été spécifiée.

### 2.3.1 Modes de fonctionnement dans IEEE 802.15.4

Le standard IEEE 802.15.4 offre deux modes de fonctionnement [15] :

- **Mode non balisé (non Beacon Enabled mode)** : Aussi appelé Unslotted mode où aucun mécanisme de duty cycle (réveil périodique) n'a été spécifié en l'état. Ce mode est plus adapté à des applications où les nœuds n'ont aucune contrainte énergétique (alimenté sur secteur). Dans ce mode il y a absence de synchronisation entre les nœuds du réseau. Afin que ces derniers puissent communiquer entre eux, ils doivent laisser leur radio allumée ou se réveiller périodiquement afin d'interroger le coordinateur pour savoir s'il y a des messages en attente, ce mode étant laissé ouvert par le standard on peut imaginer l'utilisation soit de la technique de préambule soit d'une méthode de synchronisation. Dans ce mode le mécanisme pour se protéger des collisions sera le CSMA/CA Unslotted.
- **Mode balisé (BeaconEnabled mode)** : Ou Slotted mode, étant le mode le plus important dans le standard, ses performances en terme de débit, consommation énergétique, packet delivery ratio et fiabilité ont fait l'objet d'un grand nombre d'études. Le mode Beacon Enabled est un mode dit synchronisé. Les nœuds doivent suivre une structure périodique appelée super trame, Cette structure périodique commence toujours par l'envoi d'une trame par le coordinateur appelée Beacon. Cette trame Beacon a pour rôle de synchroniser les nœuds avec leur coordinateur, et fournir toutes les informations vitales

au fonctionnement du réseau, comme par exemple l'identifiant du réseau, les données en attente au niveau du coordinateur (communication indirecte), le temps où le prochain Beacon est envoyé et la durée de la partie active de la super trame, etc.

## 2.3.2 Architecture en couches

La norme IEEE 802.15 couvre les couches : physique et la sous couche MAC de la couche liaison de données [16].

### 2.3.2.1 Couche Physique dans IEEE 802.15.4

La couche physique a pour rôle de gérer le support physique sur lequel seront faites les transmissions. Elle définit les techniques par lesquelles les bits seront transformés en signaux analogiques et inversement. La couche physique dans IEEE 802.15.4 offre les fonctionnalités suivantes :

- Gestion de l'activation et de la désactivation du module radio, la radio pourra prendre trois états différents : réception, émission et éteinte.
- La possibilité de remonter à la couche supérieure des informations sur la qualité du lien radio après réception d'une trame : LQI (Link Quality Indication).
- La détection de l'occupation ou non du médium : CCA (Clear Channel Accessment).
- La sélection d'un canal de transmission parmi les différents canaux disponibles.

### 2.3.2.2 Couche MAC dans IEEE 802.15.4

La couche MAC (Medium Access Control) dans ce standard, comme son nom l'indique, aura pour rôle de diriger l'accès au canal avec le mécanisme CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Cette couche offrira un mécanisme d'économie d'énergie, une des fonctionnalités les plus importantes dans les réseaux de capteurs sans fil. Nous pouvons résumer ce mécanisme d'une manière très simple : l'utilisation de la radio représente le plus grand facteur de dépense énergétique qui doit être limitée au strict minimum en la gardant éteinte la majorité du temps.

La couche MAC offre aussi d'autres fonctionnalités de contrôle liées à :

- Gestion du réseau.
- Gestion des acquittements.
- Gestion de l'adressage (adresse IEEE 802.15.4 codé sur 64 et 16 bits).
- Allocation et la gestion des Slots dédiés.
- Mécanisme de découverte de voisinage.

## 2.4 CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance)

Dans un réseau local Ethernet classique, la méthode d'accès utilisée par les machines est le CSMA/CD pour lequel chaque machine est libre de communiquer à n'importe quel moment. Chaque machine envoyant un message vérifie qu'aucun autre message n'a été envoyé en même temps par une autre machine. Si c'est le cas, les deux machines patientent pendant un temps aléatoire avant de recommencer à émettre.

Dans un environnement sans fil, la méthode d'accès utilisée par les machines est le CSMA/CA. Ce protocole utilise un mécanisme d'évitement de collision basé sur l'écoute du support afin de savoir s'il est libre ou pas avant l'émission, et sur le principe d'accusé de réceptions (ACK) réciproques entre l'émetteur et le récepteur pour pouvoir détecter s'il y a lieu une collision en non réception de cet ACK, si l'émetteur ne reçoit pas l'accusé de réception, alors il retransmis le fragment jusqu'à ce qu'il l'obtienne ou abandonne au bout d'un certain nombre de retransmissions [17].

### 2.4.1 Déroulement du protocole CSMA/CA

Une station voulant émettre des données commence par déterminer si le médium est libre [17] :

1. Si le médium est libre après un temps spécifique (DIFS), elle peut envoyer ses données immédiatement. Si la station reçoit un acquittement (ACK) donc l'émission réussie, sinon échec d'émission : pour la retransmission aller à 2, si elle n'a pas atteint le nombre de retransmissions permis.
2. Si le médium est occupé elle doit différer cet envoi de la manière suivante :
  - i) Attente jusqu'à ce que le médium redevienne libre.
  - ii) Puis attente d'un temps DIFS + un temps aléatoire (Backoff) de CW slots (Contention Windows, fenêtre de contention).
  - iii) Si, à la suite de cette attente, le médium est toujours libre, la station a gagné l'accès au médium et peut ainsi émettre ses données ; Si la station reçoit un acquittement (ACK) donc l'émission réussie, sinon échec d'émission : pour la retransmission aller à 2, si elle n'a pas atteint le nombre de retransmissions permis.
  - iv) Si, elle est dans la phase de fenêtre de contention, le médium est repris par une autre station, le processus est abandonné jusqu'à la prochaine période (aller à i). Cependant, la station mémorise le temps qu'il lui restait à attendre, pour reprendre là où elle était restée, lors de la prochaine négociation d'accès au médium.

3. Quand une station cible reçoit une trame, elle doit envoyer un accusé de réception à l'émetteur après avoir attendre un temps SIFS.

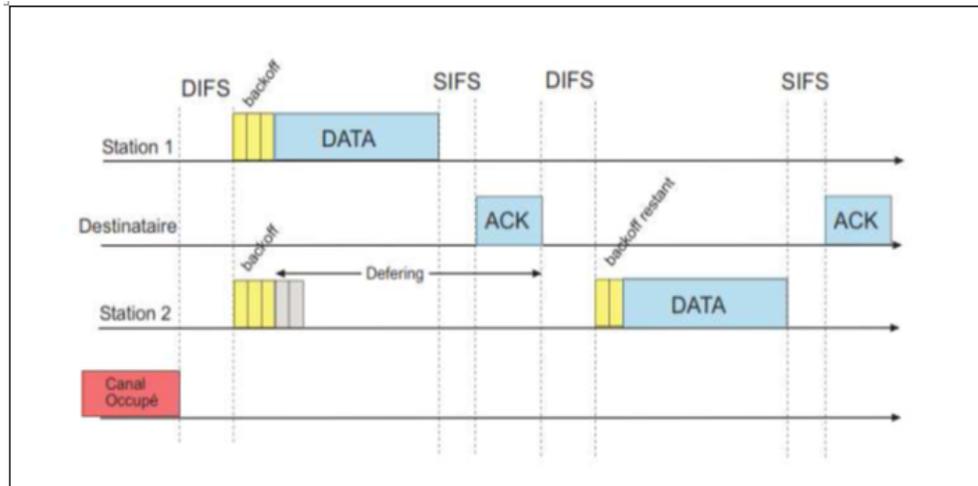


FIGURE 2.4 – Accès au médium en mode CSMA/CA [17].

### 2.4.2 Organigramme de la procédure CSMA/CA

La figure 2.5 résume le fonctionnement de la procédure CSMA/CA et de l’algorithme de Backoff.

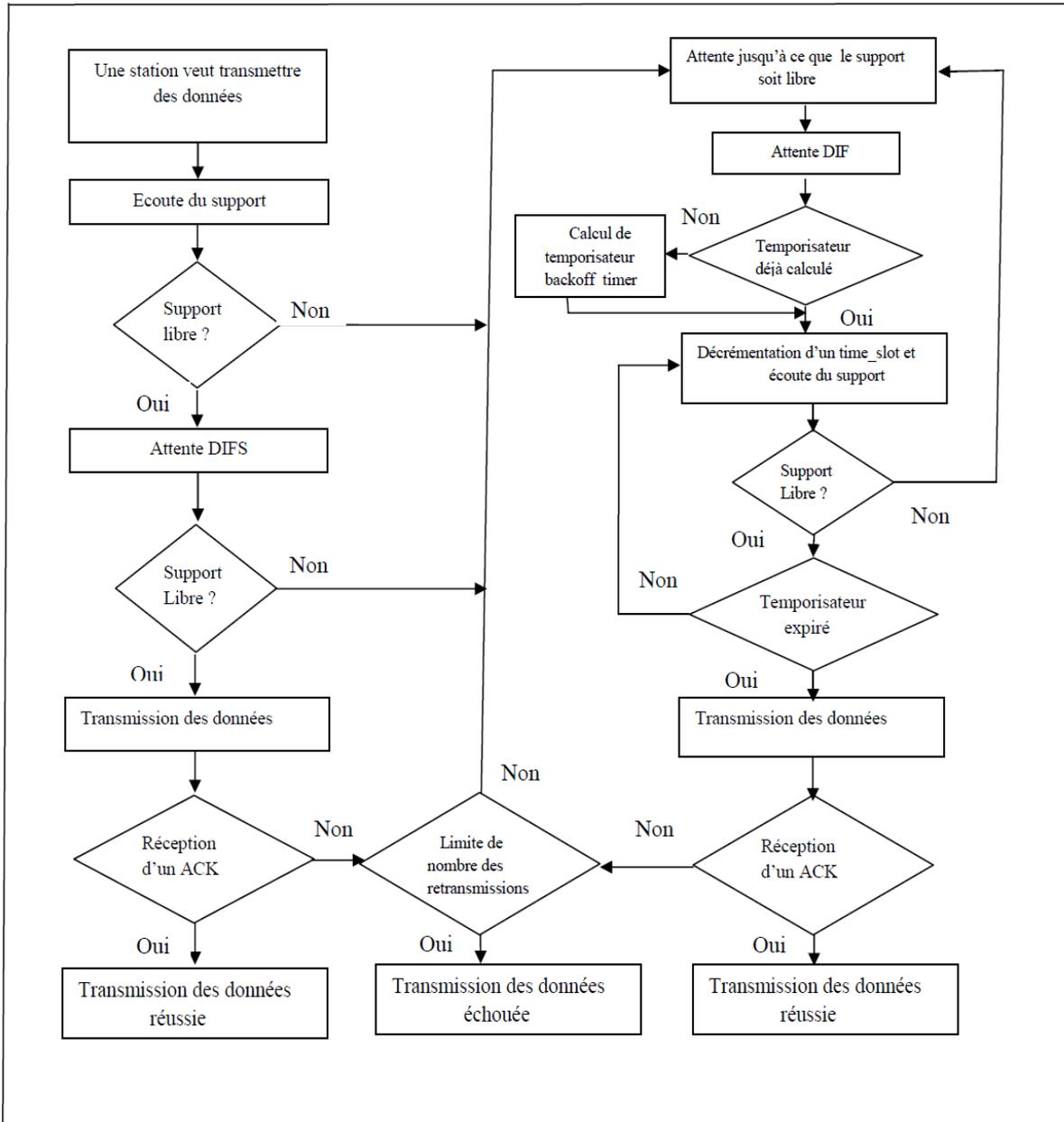


FIGURE 2.5 – la Procédure CSMA/CA

## 2.5 Méthodes d'accès au canal

La Couche MAC d'IEEE 802.15.4 propose l'utilisation du mécanisme CSMA/CA comme méthode d'accès au canal afin de se prémunir des collisions. Le CSMA/CA a pour principe de désynchroniser le temps d'envoi des trames des différents nœuds en utilisant un tirage de temps aléatoire dans un intervalle conditionné par le BackoffExponent. Après ce temps d'attente les nœuds se doivent de vérifier l'état du canal avant de transmettre [12]. Dans IEEE 802.15.4 deux versions de CSMA/CA ont été proposées fonctionnant de la même façon à quelques différences près : CSMA/CA Slotté (Slotted CSMA/CA) utilisée dans le mode Beacon, et CSMA/CA non-slotté (Unslotted CSMA/CA). Avant d'entamer la description du fonctionnement du CSMA/CA nous devons tout d'abord définir la terminologie utilisée dans IEEE 802.15.4 :

- Unit BackoffPeriod : Unité de temps égale à 20 symboles.
- BE : BackoffExponent initialisé à MacMinBE=3 par défaut.
- Backoff : Temps d'attente aléatoire tiré dans l'intervalle :  $[0, 2^{\text{BE}} - 1] * \text{UnitBackoffperiod}$ .
- CW (Contention Window) : Ce paramètre n'est utilisé que dans le mode Slotted et représente le nombre d'unités de temps après le temps Backoff où le canal doit être libre avant transmission.
- NB : Nombre d'essais d'accès au canal pour la transmission en cours, initialisé à zéro.
- macMaxCSMABackoffs : Le nombre de fois où le canal est trouvé occupé avant d'abandonner l'émission de la trame, initialisé à 3
- macMaxFrameRetries : Le nombre de retransmissions maximum autorisé pour chaque trame, initialisée à 4.

Le Fonctionnement du CSMA/CA slotted est le suivant. Avant tout essai de transmission d'une nouvelle trame les nœuds doivent effectuer les étapes suivantes :

1. Initialisation des paramètres à leur valeur par défaut (CW, NB, BE)
2. Tirage d'un temps aléatoire (Backoff) afin de désynchroniser les transmissions.
3. Les nœuds alignent leur début de Unit BackoffPeriod sur ceux de la super trame, tous les nœuds doivent s'assurer qu'il reste assez de temps avant la fin de la super trame pour le Backoff, les deux CW, l'envoi de la trame, et l'acquittement si ce dernier est demandé.
4. A la fin du temps d'attente aléatoire (Backoff) les nœuds doivent tester l'état du canal en effectuant ce qu'on appelle un CCA (Clear Channel Assessment) qui dure 8 symboles de temps :
  - Si le canal est détecté occupé, le nœud incrémente toutes les constantes de l'algorithme (NB=NB+1, BE=min (BE+1, macMaxBE) et CW=2).

- Si  $NB > macMaxCSMABackoff$  le nombre maximum d'essais permis pour la transmission d'une trame est dépassé, le nœud abandonne l'envoi de la trame en cours. Dans le cas contraire le nœud retire un nouveau Backoff a fin de retenter l'envoi de la trame (étape2).
- Si au contraire le canal est détecté libre le nœud décrémente la valeur de  $CW$  ( $CW=CW-1$ ).
- Si  $CW=0$  la trame est envoyée au début du prochain Unit BackoffPeriod, si non le nœud exécute un nouveau CCA après un Unit BackoffPeriod. Si non le Beacon Enabled mode est utilisé c'est le CSMA/CA Unslotted qui sera alors utilisé par les nœuds afin d'accéder au canal. Comme pour le CSMA/CA Slotted, les nœuds passeront par les deux premières étapes citées ci-dessus : initialisation de ( $CW$ ,  $NB$ ,  $BE$ ) et tirage du temps de backoff aléatoire. Les seules différences avec le Slotted CSMA/CA résident dans les deux points suivants :
  1. Les périodes de décrémentation des UnitBackoffPeriod entre les nœuds en concurrence pour l'accès au médium sont à l'encontre du Slotted CSMA/CA indépendantes dans le temps i.e la décrémentation du backoff commence immédiatement après le temps d'attente aléatoire sans synchronisation sur le unit-Backoff-Period puis qu'il y a absence de super trame.
  2. Avant de pouvoir transmettre, l'émetteur doit sonder le canal afin de savoir s'il est libre. En mode Unslotted ceci n'est fait qu'une seule fois, alors qu'en mode Slotted CSMA/CA le canal devra être libre durant deux UnitBackoff Period avant l'émission de la trame.

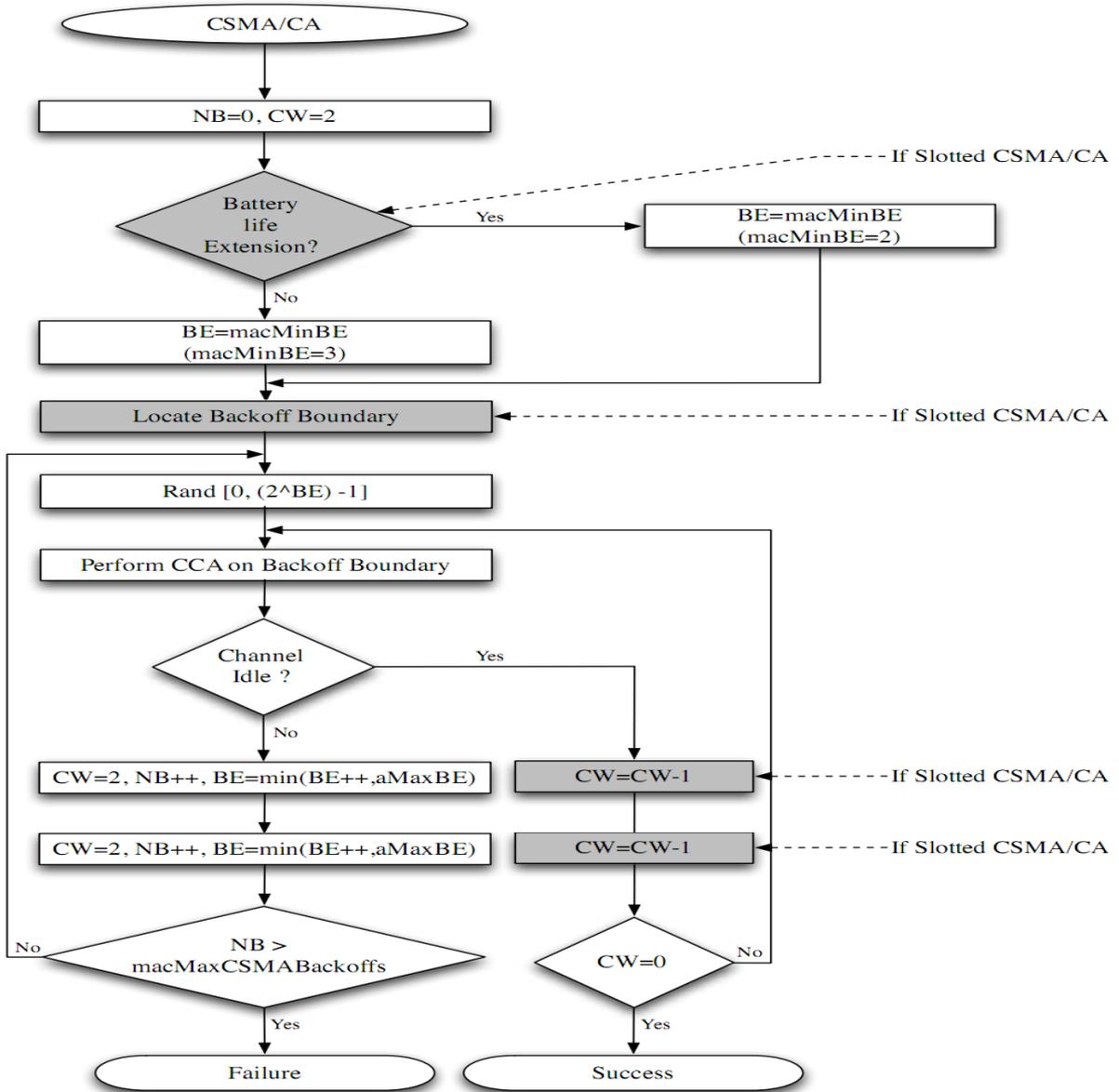


FIGURE 2.6 – CSMA/CA Slotted et Unslotted dans IEEE 802.15.4 [12].

## 2.6 Conclusion

Dans ce chapitre nous avons présenté la norme IEEE 802, nous avons défini les différents aspects des couches physiques et MAC du standard 802.15.4, la couche MAC utilise le mécanisme CSMA/CA pour partager par l'écoute, cela consiste pour une station à écouter le support pour détecter s'il y a un signal porteur et attendre, si c'est le cas, qu'il soit libre avant de transmettre. Finalement nous avons déterminé les topologies supportées dans IEEE 802.15.4 et les modes de fonctionnement dans IEEE 802.15.4 ainsi que les contraintes qui doivent être prises en compte pour éviter les collisions, cette section s'attache à exposer le fonctionnement de la méthode d'accès au canal (CSMA/CA) utilisé dans IEEE 802.15.4.

## CHAPITRE 3

# ADAPTATION DE CSMA/CA POUR LE TEMPS RÉEL

### 3.1 Introduction

Dans ce chapitre, il est question de présenter notre proposition qui consiste à rendre le protocole d'accès au support CSMA/CA sous Contiki plus adapté en temps réel.

Dans un premier temps, nous allons citer l'objectif de notre travail et détailler la problématique à l'origine de ce dernier. Dans un second temps, nous allons revenir sur le système d'exploitation Contiki - introduit au chapitre 1 - plus précisément sur la pile Rime et la couche MAC afin de présenter les modules qui sont exploités par l'application. Nous décrivons également le protocole CSMA/CA tel qu'il est implémenté sous Contiki 2.7. Nous détaillons ensuite un scénario du déroulement d'une communication. En outre, nous présentons des propositions pour remédier aux problèmes soulignés dans la problématique.

### 3.2 Problématique

Un capteur reçoit des messages à utiliser pour prendre une décision, rendre un service, activer un autre capteur...etc. dans un temps limité.

A partir des résultats des tests effectués nous avons remarqué que le mobile reçoit des messages qu'il aurait dû recevoir plutôt, ces messages retardataires engendrent plusieurs anomalies pour les systèmes temps réel (figure 3.1 ).

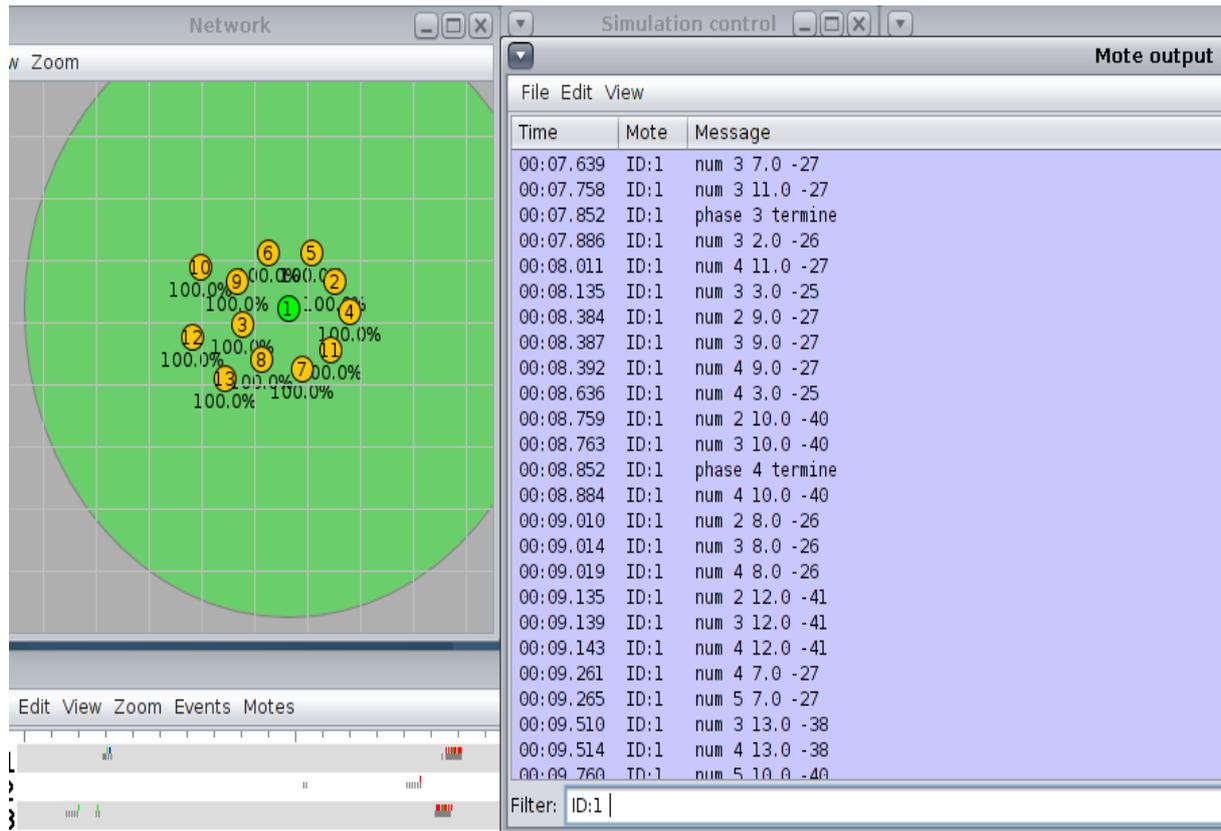


FIGURE 3.1 – Exemple de messages retardataires.

- L'information qu'ils contiennent n'est plus appropriée à l'instant présent.
- Le capteur procède à la transmission de tous les paquets en retard dans sa file d'attente, il monopolisera le canal empêchant ainsi les autres capteurs de transmettre vers le mobile les informations utiles.
- La transmission d'un message est toujours consommatrice d'énergie, alors transmettre des messages inutiles est un gaspillage d'énergie.

Dans le but de comprendre les origines de ces anomalies, nous allons d'abord étudier le système Contiki en se focalisant sur la couche MAC afin de retracer les différentes étapes sur lesquelles passe un paquet depuis sa génération jusqu'à sa transmission effective, une de ces étapes importantes est l'algorithme d'accès au support CSMA qui aussi décortiqué.

### 3.3 Objectif de notre travail

Pour voir ces messages échangés, ceux qui arrivent à temps et en retard et pour avoir un aspect général sur ce phénomène, afin de pouvoir apporter des solutions qui ont pour but d'apporter une amélioration, on déploie un ensemble de capteurs sous la topologie en étoile là où on prend le capteur du milieu comme un mobile. Ce dernier reçoit des messages de la

part des autres capteurs. A chaque fois on augmente le nombre de capteur et l'intervalle de transmission dans le but d'étudier les statistiques sur le nombre de messages transmis à temps et le nombre de messages arrivés en retard ainsi que la contrainte de consommation d'énergie.

### 3.4 Contiki en détails

Le système d'exploitation Contiki bénéficie de deux piles de protocoles pour communiquer sur le réseau, la pile UIP offre un ensemble plus que suffisant des protocoles de la suite TCP/IP pour qu'un capteur qui exécute Contiki soit capable d'obtenir une configuration IP et d'échanger des paquets de données sur internet. Mais la couche qui nous intéresse dans ce travail est la couche Rime, qui est une pile légère de transmission conçue pour les radios de basse puissance. Rime est le résultat de la superposition de plusieurs primitives chacune d'entre elles offre un service à la couche supérieure jusqu'à arriver à la couche application. Parmi ces primitives, celles qui nous intéressent [18] :

- **Anonymous best-effort local area broadcast ou abc** : Le module abc envoie des paquets anonymement (sans ajouter l'adresse de l'émetteur)
- **Best-effort local area broadcast** : Le module broadcast ajoute au paquet d'adresse Rime de l'émetteur et fait appel ensuite aux services du module abc.
- **Single-hop unicast** : Ajoute au paquet l'adresse du récepteur et fait appel ensuite aux services du module broadcast.
- **Announcements** : le module d'annonce est utilisé par les capteurs pour envoyer annonces aux voisins afin de signaler leur présence. Les annonces consistent en de petites valeurs sur 16 bits.

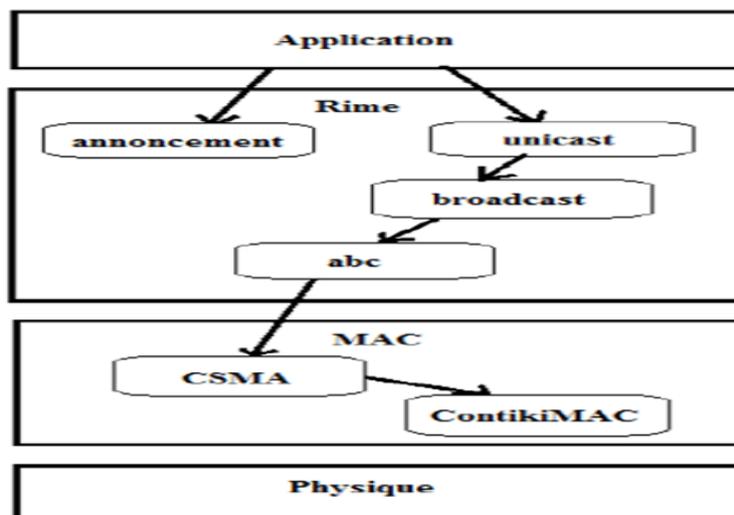


FIGURE 3.2 – Architecture de contiki.

La pile Rime utilise des canaux représentés par un nombre entier de 16 bits qui doit être le même chez l'émetteur et le récepteur pour ouvrir une connexion. Lors de l'ouverture des connexions (car entre deux capteurs on peut ouvrir plusieurs connexion) le système Contiki fait appel aux mécanismes de callback (appel de retour), en fait, les primitives supérieures passent aux modules inférieurs des pointeurs vers des fonctions qui seront appelées en retour par ces dernier, par exemple lors de la réception d'un paquet, cela permet de remonter aux couches supérieures.

Enfin, un capteur souhaitant utiliser la couche Rime doit impérativement posséder une adresse Rime qui consiste en un simple nombre.

Une fois le paquet arrivé au module abc de la part de la couche application, ce dernier veille à le passer à la couche MAC. Dans la couche MAC le paquet va être traité par deux primitives, dans un premier temps il doit passer par l'algorithme CSMA puis par Contiki MAC.

La couche MAC se situe juste en dessous de la couche Rime (voir la Figure 3.2), elle consiste en un ensemble de protocoles de conservation d'énergie comme Contiki MAC et d'accès au support comme CSMA. Les appareils sans fil de basse puissance doivent garder leurs émetteurs récepteurs par radio éteinte autant que possible pour atteindre une faible consommation d'énergie, mais ils doivent se réveiller assez souvent pour pouvoir recevoir la transmission de leurs voisins. Contiki MAC est un protocole qui permet aux nœuds de participer à la communication réseau tout en maintenant leurs radios arrêtées pour approximativement 99% du temps. Cela est possible car Contiki MAC permet au capteur de se réveiller périodiquement et d'écouter le support, si une transmission est détectée le capteur est gardé éveillé pour recevoir le prochain paquet. Quand le paquet est reçu avec succès, le récepteur envoie un accusé de réception. Les paquets qui sont envoyés en broadcast ne réclament pas d'être acquittés, au lieu de cela, l'émetteur envoie à plusieurs reprises le paquet pendant la durée de son éveil, pour s'assurer que tous les voisins l'ont reçu.

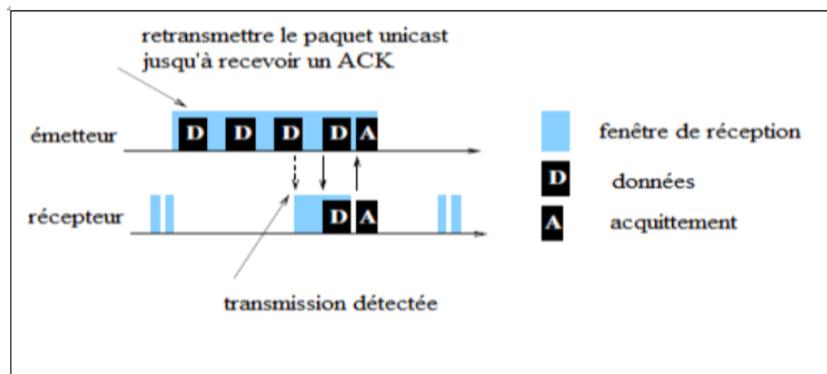


FIGURE 3.3 – Le fonctionnement de contikiMAC lors d'un unicast [18].

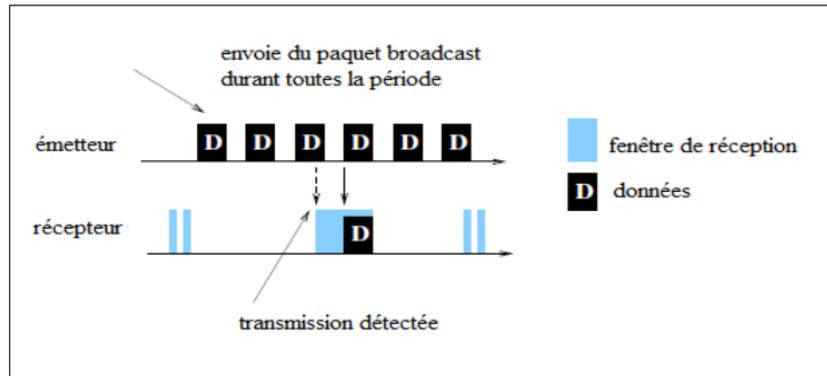


FIGURE 3.4 – Le fonctionnement de contikiMAC lors d'un broadcast [18].

### 3.4.1 Fonctionnement de CSMA/CA dans Contiki

Dans CSMA il existe principalement deux structures de données importantes, la première est la liste des voisins et la deuxième est la liste de paquets jointe à un voisin dont il est le destinataire, le nombre de voisins et de paquets varie selon la plateforme et la pile réseau utilisée (uIP ou Rime) [19].

A chaque arrivée d'un paquet à la couche MAC, CSMA vérifie si il s'agit d'un broadcast ou d'un unicast, un paquet broadcast est expédié immédiatement sans rien assigner ou allouer. Dans le cas d'un unicast, CSMA récupère l'adresse de destinataire, si ce destinataire n'existe pas dans la liste des voisins alors un espace mémoire est alloué pour l'ajouter ainsi que la liste des paquets à envoyer vers ce voisin, mais si ce voisin existe déjà alors un espace mémoire est alloué juste pour le nouveau paquet. A cette étape aussi CSMA initialise les métadonnées de ce nouveau paquet, la plus importante d'entre elle est le nombre maximum de transmissions, cette valeur est vérifiée pour la première fois dans un champ de ce nouveau paquet, en cas où elle n'est pas définie alors c'est la valeur définie par la plateforme matériel ou dans le fichier CSMA lui-même qui est prise.

Si un paquet de données nouvellement généré est le premier dans la liste vers un voisin donné, CSMA tente de l'envoyer immédiatement sans trop attendre, mais s'il ne l'est pas il sera ajouté à la fin de la liste (la liste de paquets est gérée en FIFO sauf les acquittements qui sont placés à l'en-tête de la file).

Si la transmission d'un paquet réussie, le paquet et ces métadonnées sont aussi tôt supprimés en réinitialisant les champs de la structure de ce voisin à 0 pour le prochain paquet dans la liste, si ce paquet est le dernier, dans ce cas, si tout le voisin qui est supprimé de la liste des voisins. L'envoi d'un paquet est considéré comme échoué si :

- Le capteur a réussi à le transmettre mais il n'a reçu aucun acquittement, dans ce cas, le nombre de transmissions est incrémenté.

- Une collision s’est produite en cours de transmission, dans ce cas, le nombre de collisions est incrémenté.

Si le nombre de transmission de ce paquet n’a pas dépassé le nombre maximum de transmissions fixé avant, le programme va réessayer de le retransmettre dans une durée calculée aléatoirement qui dépend de plusieurs paramètres :

$$T = \text{time} + (\text{random-rand} () \% (\text{backoff-transmissions} * \text{time}))$$

- **time** : l’intervalle de vérification du canal définie par ContikiMAC.
- **Fonction random-rand ()** : tire un nombre entier aléatoire entre 0 et 65535.
- **backoff-transmissions** : reçoit le nombre de transmission + 1, le backoff est borné par la valeur 3 afin de ne pas attendre une long durée pour tenter une prochaine retransmission.

La formule assure un temps d’attente T tel que :  $\text{time} \leq T < 4 * \text{time}$ . Un autre aspect important de CSMA dans Contiki est qu’un premier paquet peut ne pas être encore envoyé que d’autres paquets arrivent (si la longueur de la liste des paquets le permet), et lorsque le capteur obtient enfin l’accès au canal il le monopolisera jusqu’à ce qu’il envoie tous ses paquets ce qui risque de prendre un peu de temps selon le nombre de paquet dans la liste.

CSMA tel qu’il est implémenté dans Contiki 2.7 est entièrement pensé pour ne pas perdre des paquets de données même s’ils subissent des collisions ou que leurs transmissions ne sont pas acquittées un certain nombre de fois.

### 3.4.2 Quelques limitations temps réel du CSMA dans Contiki

- **Non limitation du temps de transmission** : CSMA ne fait aucune hypothèse sur l’intervalle de temps que le paquet doit rester dans la file d’attente à cause des transmissions échouées avant d’être supprimé. La durée peut être longue, et si jamais le capteur réussie à transmettre le message, ce paquet sera sûrement un retardataire.
- **Non limitation du nombre de collisions** : Dans le code source de CSMA à chaque fois qu’un paquet subit une collision, il existe une variable qui sera incrémentée mais elle n’est jamais comparée à une valeur maximale qu’elle ne doit pas dépasser, faute de quoi, le paquet sera supprimé. Si par exemple le paquet ne subit que des collisions alors il sera indéfiniment dans la liste des paquets occasionnant ainsi un très grand retard aux autres paquets.
- **Taille et Gestion de la file d’attente des paquets** : CSMA gère une file d’attente de paquets en FIFO (First In First Out), cette file d’attente est d’une taille qui varie entre 8 et 16 selon la pile réseau utilisée, cela permet donc l’accumulation de plusieurs paquets dont les premiers de la file sont probablement au retard.  
Ajouter à cela, que les paquets nouvellement créés qui ne trouvent pas de place dans la file d’attente sont supprimés et ceux dans la liste sont toujours gardés.

Ces trois éléments font que les messages qui sont générés par les capteurs et qui ne sont toujours pas transmis vers le mobile à cause des collisions, soient gardés dans la file d'attente du capteur longtemps avant d'être supprimés ou transmis au retard.

### 3.5 CSMA adapté

On prend un ensemble de capteurs sous la topologie en étoile, celui du milieu comme mobile, il va recevoir des messages de la part des autres capteurs qui sont fixés (ancres), un exemple d'établissement d'une communication entre le mobile et les ancres est donné par le diagramme de séquence de la figure 3.5 :

- Le mobile va diffuser un broadcast qui contient le message Hello par exemple à tous les capteurs ancres.
- Les capteurs ancres qui ont reçus le message du mobile vont répondre de leur part par un message, le mobile va recevoir ces messages en unicast.
- A son tour le mobile va renvoyer un acquittement pour tous les ancres lesquels il a reçu des messages, à partir de là on peut dire que la communication est établie, les capteurs ancres peuvent envoyer leurs paquets.

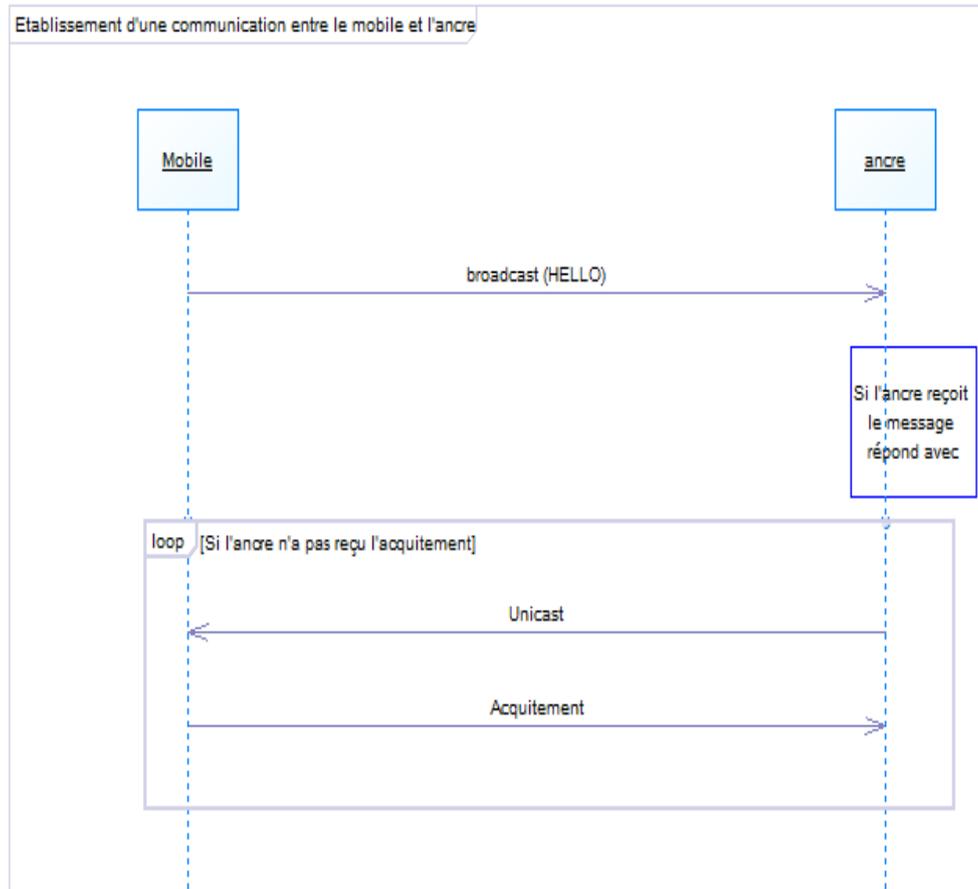


FIGURE 3.5 – Etablissement de communication entre le mobile et les ancres.

### 3.5.1 Fonctionnement

Dans cette proposition nous avons procédé d’une manière à apporter un changement sur le fonctionnement de CSMA en ce qui concerne la manière de retransmission .Au lieu de retransmettre les messages non acquittés un nombre de fois prédéfinie nous avons rendu la retransmission durant un intervalle de temps ou chaque capteur va avoir son propre instant de transmission dans cet intervalle, ce qui est illustré dans la figure 3.6

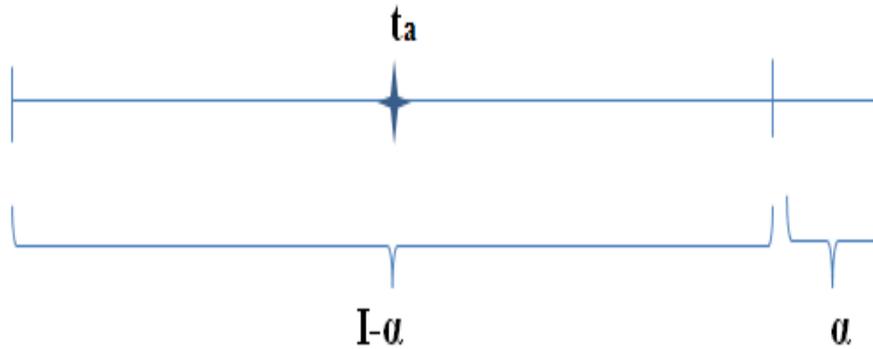


FIGURE 3.6 – Le fonctionnement de csma-adapté.

Pour aboutir à cela nous avons remplacé le fichier nommé `csma.c` de `contiki` avec un autre fichier qui contient les modifications citées ci-dessous.

- Le capteur va faire un tirage aléatoire pour trouver l'instant  $ta$  où il va transmettre,  $ta$  est inclut dans l'intervalle  $I-a$ .
- Si la transmission est réussie, càd réception d'un acquittement, le capteur va attendre jusqu'à ce que l'intervalle  $I-ta$  s'écoule pour transmettre un nouveau paquet.
- Sinon, il va retransmettre durant l'intervalle  $I-ta$  jusqu'à ce qu'il reçoit un acquittement.
- Si l'intervalle est écoulé et il n'a pas reçus d'acquittement, il libère le canal et élimine le paquet.

### 3.6 Conclusion

Durant ce chapitre nous avons présenté l'objectif de notre travail et la problématique qui nous a poussé à le réaliser puis le protocole CSMA et son fonctionnement dans `contiki` et ces limitations, après nous avons présenté notre proposition `csma-adpté` en décrivant les changements apporter afin de l'appliquer en déroulant des simulations ce qui va être fait dans le prochain chapitre.

## 4.1 Introduction

Nous allons consacrer ce chapitre pour le déroulement des simulations. Avant d'effectuer divers simulations portant sur chacune des propositions , nous présentons le simulateur cooja, en suite , pour chaque simulation nous récupérons les résultats des messages reçus et la consommation d'énergie par les capteurs et nous les modélisons sous forme de graphes, afin de pouvoir comparer entre les proposition en terme d'énergie et d'élimination de messages retardataires , dans le but de choisir la solution csma la plus efficace, nous commençons par tester le fonctionnement de csma tel qu'il est implémenté dans Contiki sans aucune modification puis, nous déroulons les simulation des propositions csma-adapté et sans acquittement.

## 4.2 Simulateur Cooja

Pour dérouler les tests de simulations nous avons utilisé le simulateur inclut dans le système contiki appelé cooja qui a les caractéristiques suivantes [20] :

- COOJA combine des simulations de capteur matériel de nœud et simulation du comportement de haut niveau en une seule simulation.
- COOJA est flexible et extensible en ce que tous les niveaux du système peuvent être modifiés ou remplacés.
- COOJA est une application Java, toutes les interactions avec Code Contiki se fait à travers Java Native Interface (JNI).

L'interface de simulateur cooja est composée de plusieurs fenêtres (plugins) ce qui est illustré dans la Figure 4.1

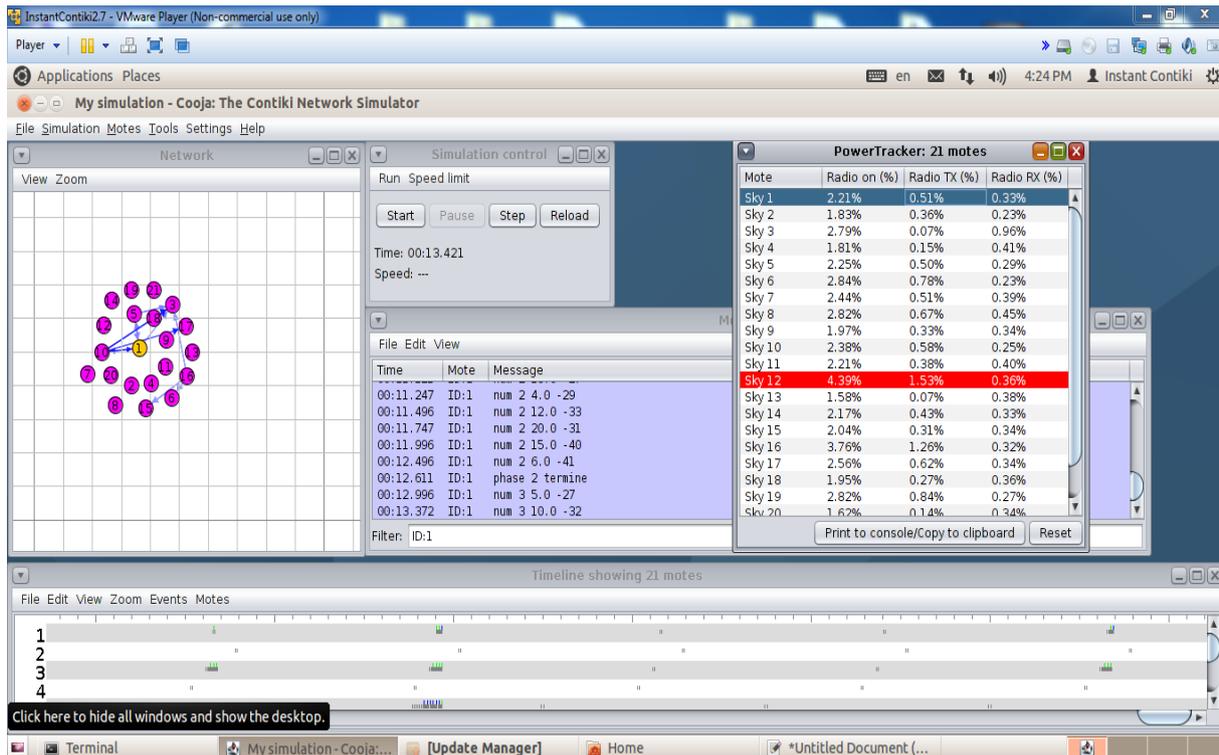


FIGURE 4.1 – Interface de simulateur cooja.

- **Mote output** : présente ce que les capteurs génèrent comme sortie via leurs ports séries.
- **Network** : présente le réseau simulé et le flux de communication durant la simulation.
- **Simulation control** : cette fenêtre contient quatre boutons :
  - 1) Start : pour démarrer une simulation.
  - 2) Pause : pour arrêter la simulation.
  - 3) Reload : pour recharger une simulation.
  - 4) Step : pour régler la vitesse de la simulation.
- **Power tracker** : pour voir la consommation d'énergie par les capteurs durant la simulation.

### 4.3 Déroulement des tests

Nous avons effectué les tests sur quatre réseaux qui ont la topologie en étoile et à chaque fois on augmente la taille du réseau en ajoutant des capteurs (10, 20, 30 et 50 capteurs) et à chaque fois on augmente l'intervalle de temps de régénération. Grâce à un programme écrit en langage C, que nous avons réalisé, nous récupérons les résultats, qui nous intéressent, suivants :

- **Moyenne de messages reçus à temps** : Si les capteurs ancrés envoient un message chaque 1s par exemple, en 1 minute (le temps de la simulation) chaque capteur va envoyer

environ 60 messages. Ce qui fait que le capteur mobile peut recevoir jusqu'à 60 groupes de messages, les messages de chaque groupe ont le même identifiant qui sera incrémenté lors de la transmission du prochain groupe.

- **Moyenne de messages arrivés en retard** : Un message dit qu'il est en retard, il n'est pas arrivé avec son groupe, nous les identifions par leurs numéros qui sont inférieurs à celui du groupe dans lequel ils sont arrivés.
- **Consommation d'énergie** : Grâce au plugin Power tracker de cooja, nous pouvons récupérer le pourcentage de consommation d'énergie de chaque capteur durant la simulation.

## 4.4 Cdma-contiki

En premier lieu, nous avons procédé à tester le fonctionnement du protocole CSMA tel qu'il est dans Contiki sans aucun changement qu'on va appeler par `cdma-contiki` par la suite. Les résultats de ce test sont représentés dans le tableau 4.1

Nombre de capteurs	intervalle de régénération (en seconde)	Moy. de msg à temps	Moy. de msg en retard	Taux en (%)		Consommation d'énergie (en %)
				A temps	En retard	
1+10	1	4.38	4.47	43.83	44.67	3.22
1+20	2	4.24	6.17	41.20	20.86	2.96
1+30	3	12.58	6.47	41.97	21.59	2.8
1+50	7	25.97	17.87	47.22	19.49	2.4

TABLE 4.1 – Résultats de test sur `cdma-contiki`.

### 4.4.1 Analyse des résultats

- **Moyenne de messages reçus à temps et en retard** : Nous avons remarqué qu'avec un intervalle de régénération et un nombre de capteurs réduit la moyenne de messages reçus à temps et celui des messages retardataires sont presque égaux. Puis avec l'augmentation de l'intervalle et le nombre de capteur, nous avons enregistré une augmentation de la moyenne des messages à temps et une diminution de celle des messages retardataires mais cette diminution n'est pas intéressante car le taux des messages retardataires reste toujours important (environ de 20%), ce qu'on peut voir sur le graphe de la figure 4.2.

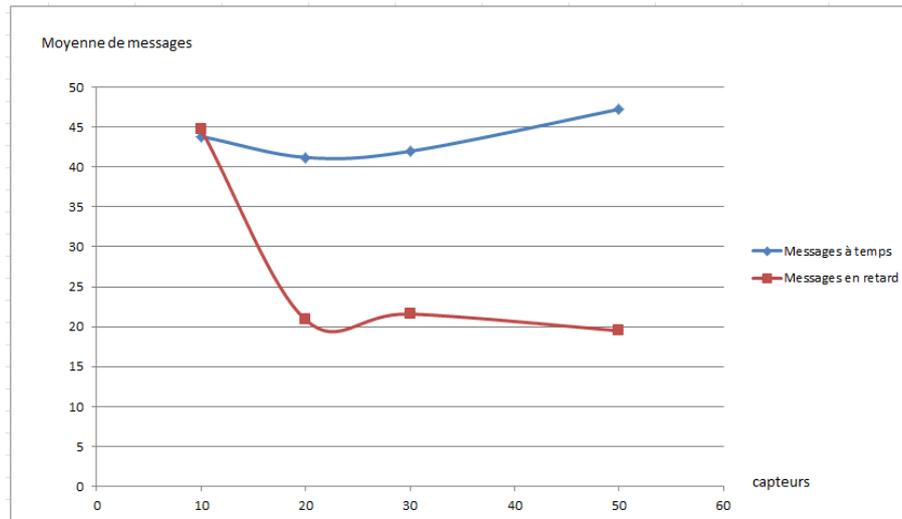


FIGURE 4.2 – Messages reçus à temps et en retard avec csma-contiki.

- **Consommation d'énergie :** Nous remarquons que quand l'intervalle de la régénération est petit, les capteurs consomment plus d'énergie, plus l'intervalle de la régénération augmente, plus la consommation d'énergie est moins importante (figure 4.3).

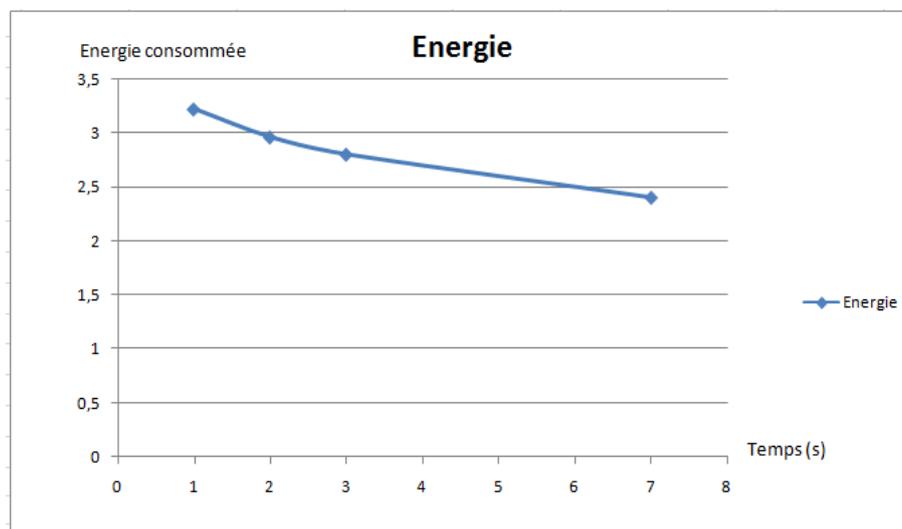


FIGURE 4.3 – Consommation d'énergie avec csma-contiki.

#### 4.4.2 Interprétation des résultats

- **Moyenne de message transmis à temps et en retard :** Avec l'augmentation de l'intervalle de régénération, le nombre de messages arrivés à temps augmentent, et celui des messages en retard assez grand car les paquets subissent beaucoup de collisions. Un

paquet dont la transmission a échoué par exemple à l'instant  $t_0$ , CSMA va le retransmettre à un instant  $t_1 = t_0 + T$  ( $T$  est donné dans la description de CSMA), si la transmission échoue plusieurs fois aux instants  $t_1$ ,  $t_2$  et  $t_3$ , CSMA permet toujours de reporter la retransmission à  $t_4 = t_3 + T$ , sauf que l'instant  $t_4$  risque de dépasser le temps de la régénération, très petit ici, et le paquet va donc arriver en retard. Un capteur peut aussi ne pas avoir accès au canal pendant longtemps à cause de trafic important dans le réseau et CSMA garde ce paquet toujours dans sa file d'attente et lorsqu'il a enfin accès au canal, ce message arrive au mobile très tardivement. Plus le temps de la régénération grandit plus les capteurs disposent d'un plus grand intervalle de temps pour retransmettre les paquets qui ont subi des collisions ou des transmissions non acquittées ou d'accéder tout simplement au canal avant que les paquets de prochain groupe ne soient générés.

- **Consommation d'énergie** : Un temps de régénération trop petit conduit à la transmission d'un grand nombre de messages, en plus des retransmissions des paquets dont les envois précédents ont échoué, d'où la consommation importante de l'énergie. Plus le temps de régénération augmente plus le nombre d'erreurs dans le réseau et de messages générés par capteur diminue, ainsi, l'activité du module radio se trouve réduite et le capteur consomme moins d'énergie.

### 4.4.3 Evaluation

Pour résumer les résultats des tests, nous pouvons dire qu'un temps de régénération petit par rapport au nombre de capteurs dans le réseau conduit à des collisions et puisque CSMA persiste à retransmettre ces paquets cela cause des messages retardataires. Il est clair que plus le nombre de capteurs est grand, plus il faut un temps de régénération plus important pour atteindre un bon résultat, mais même avec un grand intervalle du temps nous n'arrivons toujours pas à atteindre des résultats meilleurs dans ce qui concerne l'augmentation de taux de messages à temps et la réduction de celui des messages en retard.

Il faut donc trouver une solution pour réduire le nombre de messages retardataires et augmenter par la même occasion le nombre de messages à temps. Pour résoudre ce problème nous avons présenté une solution dans le chapitre III nommée CSMA-adapté. Les simulations portant sur cette proposition sont données dans la section suivante.

## 4.5 CSMA-adapté

De la même manière que dans csma-contiki, on prend un réseau avec le mobile au milieu et les ancres autour de lui et à chaque fois on augmente le nombre de capteurs et l'intervalle de régénération, les résultats obtenus sont représentés dans le tableau 4.2

Nombre de capteurs	intervalle de régénération (en seconde)	Moy. de msg à temps	Moy. de msg en retard	Taux en (%)		Consommation d'énergie (en %)
				A temps	En retard	
1+10	1	4.63	0.3	46.33	3	2.44
1+20	2	10.16	0.42	50.82	2.13	2.88
1+30	3	15.32	0.37	57.96	1.4	2.90
1+50	7	36.66	0.46	66.65	0.83	2.51

TABLE 4.2 – Résultats de test sur csma-adapté.

### 4.5.1 Analyse des résultats

- **Moyenne des messages arrivés à temps et en retard** : Nous avons enregistré une augmentation du taux de messages à temps avec l'augmentation de l'intervalle de régénération et le nombre de capteurs, le contraire avec le nombre des messages retardataires qui converge vers une valeur nulle ce qu'on peut voir sur le graphe de la figure 4.4

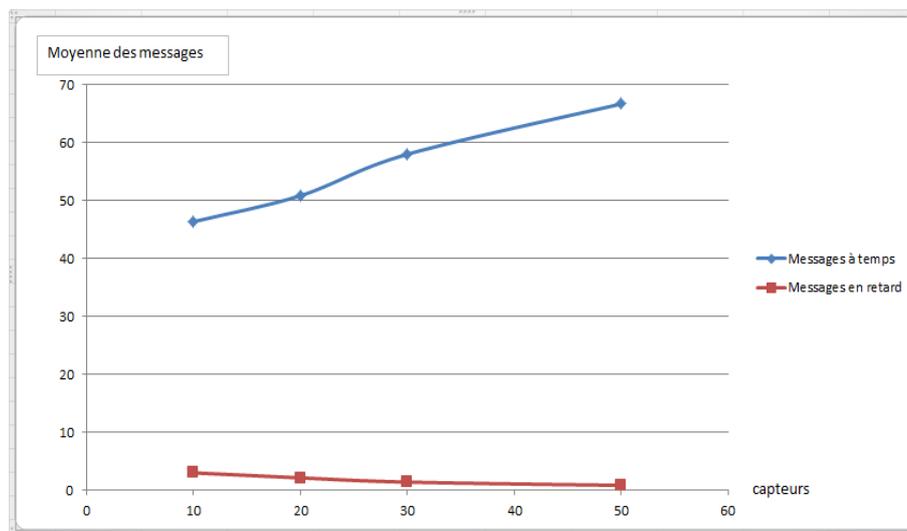


FIGURE 4.4 – Messages à temps et en retard par csma-adapté.

- **Consommation d'énergie** : Nous avons remarqué une légère augmentation de la consommation d'énergie avec l'augmentation du nombre de capteurs, mais avec un intervalle assez grand nous avons enregistré une diminution de la consommation même avec un nombre de capteur élevé (figure 4.5).

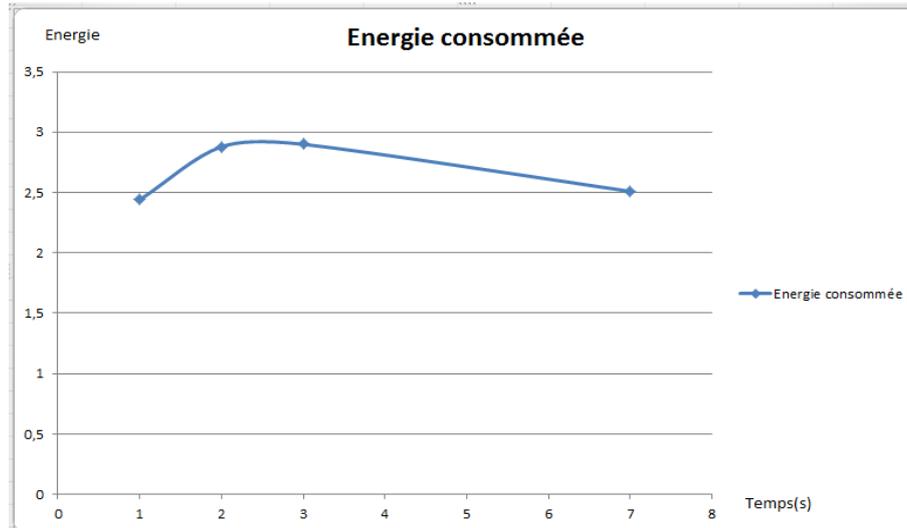


FIGURE 4.5 – Consommation d'énergie avec csma-adapté.

### 4.5.2 Interprétation des résultats

- **Moyenne de messages arrivés à temps et en retard** : Nous avons enregistré un taux de messages à temps plus élevé et un taux de messages retardataires presque nul et cela est dû à la manière de retransmission adaptée où, nous avons alloué le canal durant un intervalle pour tous les capteurs où ils vont transmettre et retransmettre leurs paquets s'ils ne sont pas reçus par le mobile. Durant cet intervalle, chaque capteur a son propre instant de transmission généré aléatoirement ce qui réduit les collisions, élimine les paquets au-delà de l'intervalle, en conséquence diminue le nombre de messages en retard tel que donné dans la figure 3.6
- **Consommation d'énergie** : La nouvelle manière de retransmission que nous avons adaptée où chaque capteur va transmettre durant un intervalle, par le fait que la retransmission dépend de cet l'intervalle ce qui augmente le nombre de retransmission par chaque capteur, ce qui rend la consommation importante.

### 4.5.3 Evaluation

Les résultats donnés par cette solution sont meilleurs que les résultats de csma-contiki, car nous avons enregistré un taux de messages à temps plus élevé et un taux de messages retardataires réduit, même si avec une consommation d'énergie un peu plus élevée, par contre avec csma-contiki nous avons enregistré un taux de messages en retard important avec une grande consommation d'énergie ce qui n'est pas intéressant.

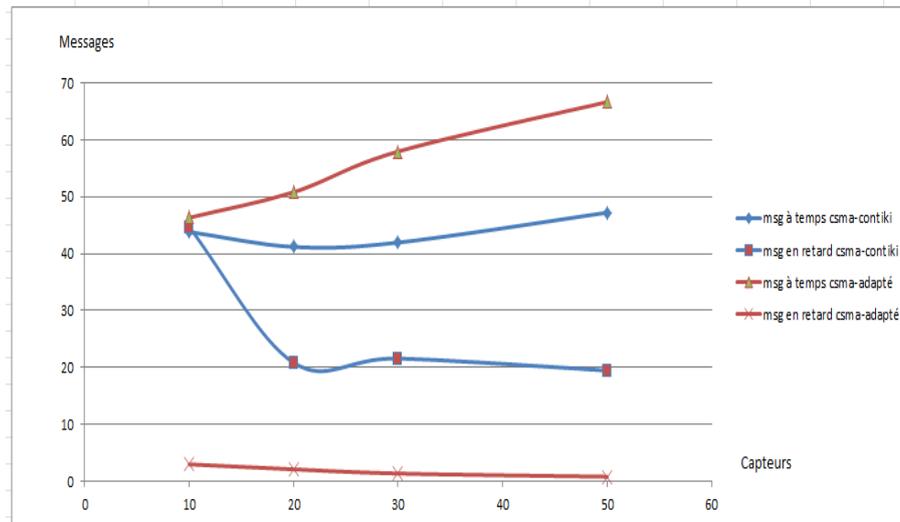


FIGURE 4.6 – Comparaison des messages à temps et en retard entre csma-contiki /csma-adapté.

En conséquence pour réduire la consommation d'énergie nous avons pensé à éviter complètement les retransmissions des messages. Pour cela contiki permet un certain nombre de configuration sur la couche MAC. C'est ce que nous allons voir dans la section suivante.

## 4.6 Transmission sans acquittement

L'objectif de cette proposition est d'annuler toute retransmission des messages et cela par l'annulation de la fonction de retransmission implémenté dans la couche MAC, pour réaliser cela nous avons introduit un fichier en-tête qu'on a appelé `project-conf.h` dans le répertoire de notre projet, ce fichier a comme contenu ce qui suit :

```
# define NETSTACK_CONF_MAC nullmac_driver
```

Lors de la compilation du projet, le fichier `PROJECT-CONF.h` est pris en compte dans le fichier `Makfile` comme étant un flag.

Pour les tests, nous avons procédé de la même manière que les tests effectués précédemment, où nous avons récupéré aussi la moyenne des messages arrivé à temps, la moyenne des messages en retard et la consommation d'énergie qui figurent dans le tableau 4.3

Nombre de capteurs	intervalle de régénération (en seconde)	Moy. de msg à temps	Moy. de msg en retard	Taux en (%)		Consommation d'énergie (en %)
				A temps	En retard	
1+10	1	3.97	1.23	39.67	12.33	1.96
1+20	2	9.39	1.03	46.97	5.15	1.58
1+30	3	15.33	0.93	51.11	3.11	1.25
1+50	7	31.35	0.58	57	1.06	1.09

TABLE 4.3 – Résultats de test sur transmission non acquittée.

#### 4.6.1 Analyse des résultats

- **Moyenne des messages arrivés à temps et en retard** : Nous avons remarqué que le taux de messages arrivés à temps augmente avec l'augmentation du nombre de capteur et l'intervalle de régénération, contrairement avec le taux de messages arrivés en retard nous avons enregistré une diminution des messages retardataires avec l'augmentation de l'intervalle de régénération et le nombre de capteurs, ce qu'on peut voir sur le graphe de la figure 4.7 :

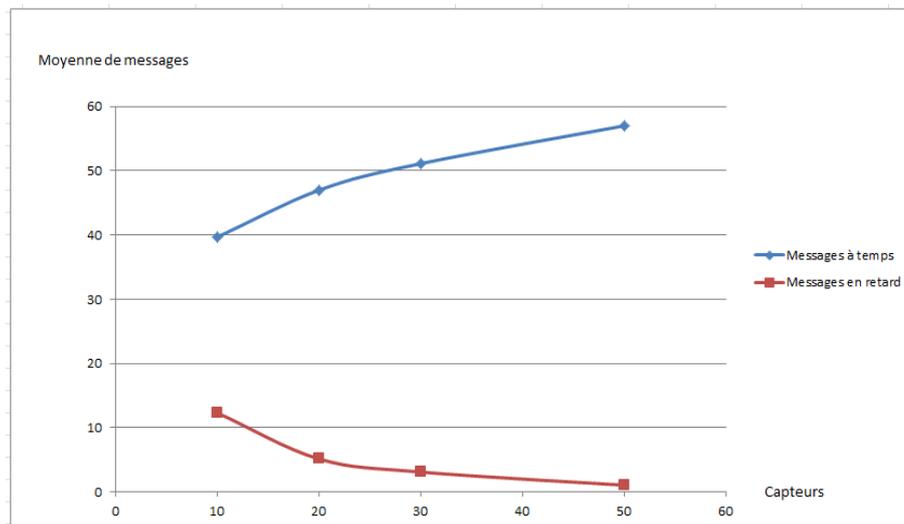


FIGURE 4.7 – Messages à temps et en retard sans ack.

- **Consommation d'énergie** : Nous avons enregistré une consommation d'énergie plus élevée par les capteurs dans les réseaux avec un intervalle de régénération plus petit et peu de capteurs contrairement aux réseaux qui ont un nombre de capteurs et un intervalle élevés (figure4.8).

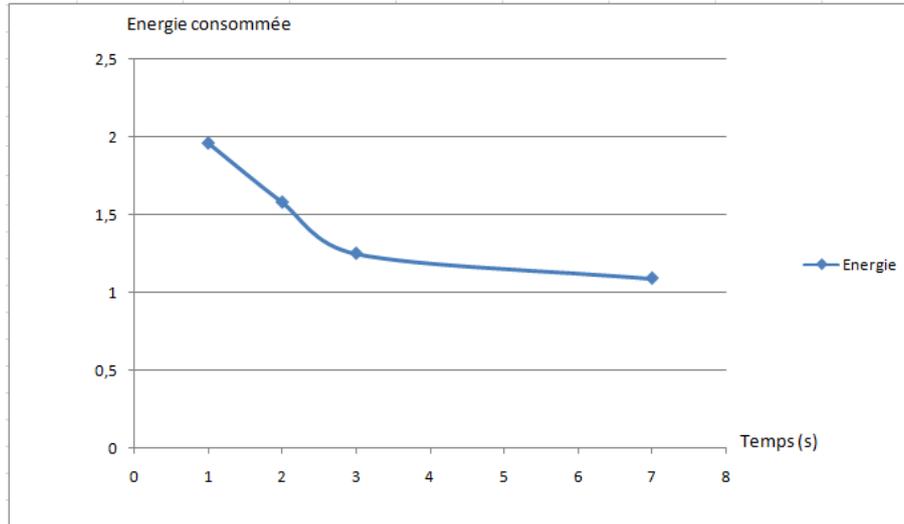


FIGURE 4.8 – Consommation d'énergie avec transmission non acquittée.

#### 4.6.2 Interprétation des résultats

- **Moyenne des messages arrivés à temps et en retard :** L'augmentation du taux de messages à temps et la diminution du taux de messages en retard due à l'augmentation de l'intervalle de génération car avec un intervalle assez grand les capteurs ont le temps pour transmettre leurs messages et avec l'absence des acquittements qui a par conséquent une absence de retransmission et une diminution des collisions, ce qui rend le nombre de messages en retard petit.
- **Consommation d'énergie :** L'absence des acquittements conduit à l'absence de retransmission ce qui évite la grande consommation de l'énergie par les capteurs.

#### 4.6.3 Evaluation

Les résultats donnés par cette solution sont meilleurs que les résultats de notre première solution (csma-adapté) et de csma-contiki, car nous avons pu aboutir à un taux de messages à temps élevé et de réduire le taux de messages retardataires avec une consommation d'énergie réduite. Dans la première proposition nous avons pu atteindre des bons résultats en ce qui concerne la diminution de taux des messages en retard mais pas en ce qui concerne la contrainte d'énergie, mais avec csma-contiki nous avons enregistré un taux de messages en retard important avec une grande consommation d'énergie ce qui n'est pas intéressant.

## 4.7 Comparaison selon l'intervalle de régénération

Dans le but de savoir l'influence de l'intervalle de régénération sur l'augmentation du taux des messages à temps et la bonne exploitation des capteurs sur le plan énergétique, nous avons procédé de la manière suivante : Nous avons pris le réseau à 20 capteurs et à chaque fois nous augmentons l'intervalle de régénération de 1s à 2s puis 3s, en gardant toujours le nombre de capteurs fixé à 20, nous avons fait la même chose pour tous les cas (csma-contiki, csma-adapté et transmission sans ack), les résultats sont présentés dans le tableau 4.4

Cas de test	intervalle de régénération (en seconde)	Moy. de msg à temps	Moy. de msg en retard	Taux en (%)		Consommation d'énergie (en %)
				A temps	En retard	
Csma de contiki	1	1.62	10.38	8.53	54.63	4
	2	4.24	6.17	41.20	20.86	2.96
	3	7.91	3.84	45.92	18.35	2.23
Csma adapté	1	3.27	0.4	16.33	2.00	3.25
	2	10.16	0.42	50.82	2.13	2.88
	3	10.93	0.53	54.67	2.67	1.15
Sans Ack	1	10.93	0.53	15.56	5.93	2.06
	2	9.39	1.03	46.97	5.15	1.58
	3	9.37	0.57	49.3	2.98	1.5

TABLE 4.4 – Résultats de test avec variation d'intervalle.

D'après les résultats des tests, nous avons remarqué que plus qu'on augmente l'intervalle de régénération le taux de messages reçus à temps par le mobile augmente pour tous les cas, mais avec la proposition csma-adapté que nous avons enregistré un taux très élevé (54.67) avec un intervalle de 3s, et en ce qui concerne la consommation d'énergie nous avons remarqué que plus que l'intervalle de régénération augmente les capteurs consomment moins d'énergie, et avec 3s nous avons remarqué que csma-adapté est meilleur sur le plan énergétique et la moyenne des messages à temps, là nous avons enregistré avec csma-adapté une consommation d'énergie de 1.15 unités et un taux de messages à temps de 54.67, par contre avec transmission non acquittée, 49.3 messages à temps avec 1.5 unités d'énergie consommées.

Donc, on peut dire que l'intervalle de régénération a une grande influence soit sur la réduction des collisions ou sur la bonne exploitation des capteurs sur le plan énergétique.

## 4.8 Evaluation finale

D'après les résultats des tests que nous avons effectués sur nos deux propositions et csma-contiki, nous avons remarqué que la première proposition (csma-adapté) est la meilleure en ce qui concerne la l'élimination de retard et l'augmentation de nombre des messages transmis à temps et sur le plan énergétique la deuxième proposition est meilleure car elle garantit une bonne exploitation des capteurs et économise l'énergie, ce qu'on peut voir sur les graphes des figures 4.9 et 4.10 :

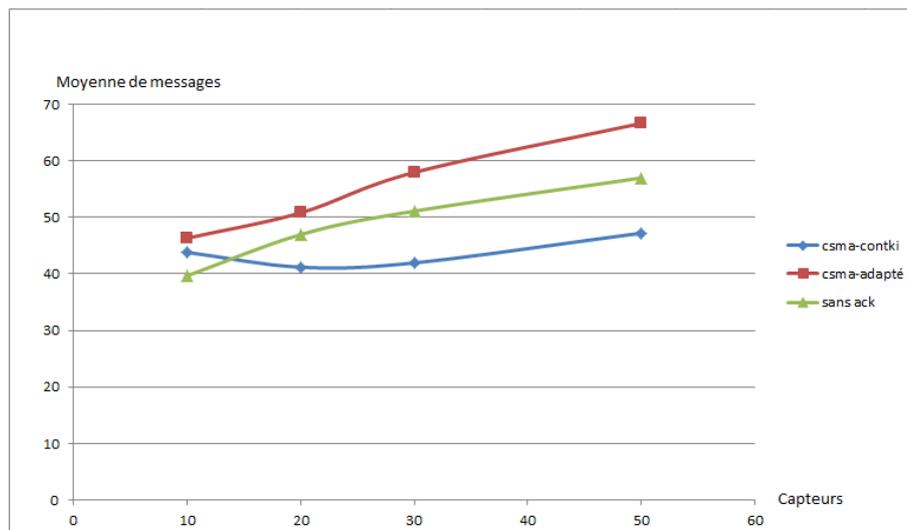


FIGURE 4.9 – Comparaison des messages à temps entre csma-contiki /csma-adapté /sans ack.

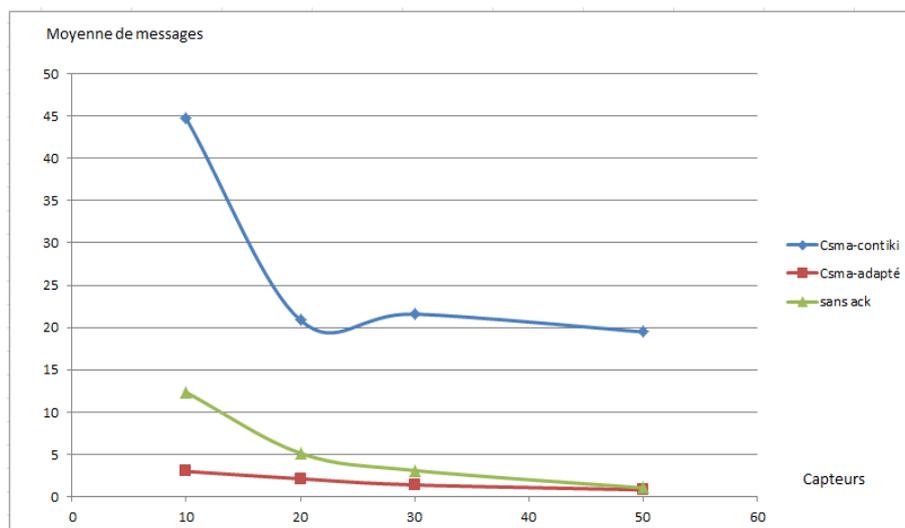


FIGURE 4.10 – Comparaison des messages retardataires entre csma-contiki /csma-adapté /sans ack.

Sur le graphe 4.9, on peut clairement voir qu'avec csma-adapté la moyenne de messages transmis à temps est plus grand par rapport aux autres modes de transmission, elle approche de 40 messages par contre avec les modes sans ack et csma-contiki, elle est à 31 et 26 messages respectivement. Pour les messages retardataires, c'est avec csma-adapté que nous avons enregistré le moins de retard, en deuxième position on trouve sans ack puis csma-contiki qui a enregistrée une moyenne de messages retardataires importante ce qu'on peut voir sur la figure 4.10

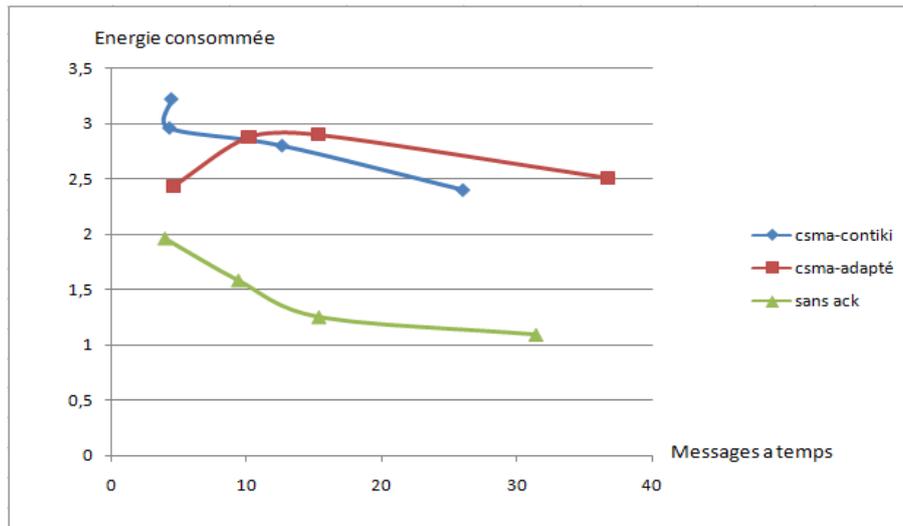


FIGURE 4.11 – Comparaison de la consommation d'énergie entre csma-contiki /csma-adapté /sans ack.

D'après le graphe de la figure 4.11 on peut constater que la deuxième proposition est la meilleur sur le plan énergétique là on peut voir que les capteurs ont consommé le moins d'énergie, à la deuxième position, on a csma-adapté qui a enregistré des résultats meilleurs que csma-contiki, car comme on peut distinguer cette différence sur le graphe, par exemple avec csma-adapté les capteurs ont consommé 2.5 unités pour atteindre une moyenne de 40 messages transmis à temps, par contre avec la même valeur d'énergie consommée (2.5) unités csma-contiki n' a pu atteindre que 23 messages à temps.

Pour conclure, nous avons pu apporter une amélioration à contiki soit sur le plan énergétique ou en ce qui concerne l'élimination du retard et cela avec nos deux proposition (transmission non acquittée, csma-adapté) respectivement.

## 4.9 Conclusion

Nous avons consacré ce chapitre pour le déroulement des simulations, en testant d'abord csma-contiki, puis nos deux propositions csma-adapté et sans acquittement, et nous avons fait une comparaison entre les résultats récupérés, afin de faire une évaluation qui est la suivante : Avec la première proposition nous avons pu atteindre une très grande réduction des messages retardataires et la deuxième est venue dans le but d'économiser de l'énergie. Dans la pratique le choix d'une des propositions dépend du domaine de l'application. Si la contrainte d'énergie est très forte, de préférence, il faut opter pour la solution sans acquittement au détriment du taux de messages reçus à temps. Dans le cas contraire, si on s'intéresse plus au nombre de messages reçus à temps, pour une prise de décision par exemple, on opte pour la solution csma-adapté en consommant un peu plus d'énergie.

## CONCLUSION GÉNÉRALE

Les réseaux de capteurs sans fil constituent des sujets de recherche innovants pour diverses disciplines des sciences et techniques de l'information et de la communication, mais avec toutefois des contraintes spécifiques s'élevant en défis. Parmi les problèmes posés dans ce type de réseaux, on trouve :

- La réception des messages retardataires : causé par des collisions sur les supports de transmission.
- La contrainte énergétique : il y a des cas où on ne peut pas recharger les batteries des capteurs, alors il est important de bien exploiter cette énergie.

Notre travail a pour but d'apporter une amélioration sur ces deux plans, pour cela nous avons proposé dans ce mémoire une adaptation du protocole d'accès au support CSMA/CA pour une application temps réel.

Notre travail fait l'objet d'assurer une meilleure exploitation des capteurs en terme de quantité d'informations, la validité des messages (éliminer les paquets retardataire) et garantir un temps de réponse souhaité. Pour cela nous avons procédé en deux phases, dont nous avons commencé par la description de la problématique du csma/ca-Contiki, ensuite nous avons développé deux solutions.

- La première solution : consiste à intégrer un contrôle de la validité des paquets par un intervalle de temps bien déterminé dans le corps du CSMA, c'est-à-dire que CSMA en quelque sorte lance un temporisateur pour chaque paquet prêt à être transmis. Si le capteur a réussi l'envoi de ce paquet avant l'écoulement de ce temporisateur, donc le message arrivera au mobile à temps, sinon le paquet sera supprimé pour qu'il ne soit pas en retard, ce qui assure l'élimination des retardataires, une meilleure exploitation des capteurs, réduction du nombre de collision en gérant la compétition des capteurs et ainsi une économie d'énergie.
- La deuxième solution : nous avons pensé à éliminer la retransmission à fin d'éliminer

les messages qui seront probablement retardés en supprimant ces derniers dès leurs premier échec de transmission (non réception de l'ACK). En résultat, nous avons réduit le phénomène des retardataires, et nous avons pu gagner de l'énergie car avec l'absence de retransmission, les capteurs ne consomment pas beaucoup d'énergie.

Nous avons eu des bons résultats sur le plan énergétique avec la première solution et en ce qui concerne la réduction des messages retardataires avec la deuxième solution, alors ces deux solutions sont toutes les deux utiles selon le besoin, c'est à dire si la contrainte énergétique est forte, on va opter pour la deuxième solution, si on ne s'intéresse pas au côté énergétique, ce qu'on veut c'est d'avoir plus d'information au moment du besoin mais pas en retard, pour la prise d'une décision par exemple, on va opter pour la première solution (csma-adapté).

Comme perspectives, nous souhaitons prochainement d'utiliser d'autre simulateur par exemple, afin de pouvoir augmenter le nombre de capteur le plus possible (200, 400,...), et de pouvoir exécuter ces programmes sur des capteurs réels. Ce travail nous a permis d'acquérir plusieurs connaissances de bases sur le fonctionnement des réseaux de capteurs sans fil, en générale et sur le système d'exploitation Contiki en particulier. Nous avons également appris à simuler des réseaux de capteurs sous le simulateur Cooja et de faire exécuter aux capteurs les applications de notre choix.

## BIBLIOGRAPHIE

- [1] J.P ARNAUD. *Réseaux et Télécoms*. DUNOD , ISBN 2100079867, 2003.
- [2] J.F PILLOU. *Tout sur les réseaux et Internet*. DUNOD,2ème édition, 2009.
- [3] S. KUMAR C. CHONG. *Sensor Networks : Evolution, Opportunities, and Challenges*. 2010.
- [4] G. ASCH. *Acquisition de données du capteur à l'ordinateur*. ISBN : 2100063103.
- [5] Z. BOUZIDI A. BENAMEUR. *Mise en place d'un réseau de capteurs sans fil pour l'irrigation intelligente*. mémoire de master uiversité Abou Bakr Belkaid Tlemcnen, juin 2012.
- [6] J. BEAUDAUX. *Partitionnement logique dans les réseaux de capteurs sans fil*. Université de Strasbourg, 2010.
- [7] D. Eya. *Routage avec QoS temps réel dans les réseaux de capteurs*. Projet de fin d'études en Ingénieur en Télécommunications, 2007.
- [8] D.JANAKIRAM A. MALLIKARJUNA and G. KUMAR. *Operating Systems for Wireless Sensor Networks : A Survey Technical Report*. Indian Institute of Technology Madras, 2007.
- [9] P. FICHEUX. *Linux embarqué*. ISBN : 2-212-11674-8, 2005.
- [10] H. DAI J. DENG J. ROSE A. SHETH B. SHUCKER C. GRUENWALD A. TORGERSON S. BHATTI, J. CARLSON and R. HAN. *MANTIS OS : An Embedded Multithreaded Operating System for Wireless MicroSensor Platforms*. University of Colorado at Boulder, 2005.
- [11] M. DIAS L. IANNONE, F. BENBADIS and S. FDIDA. *Some applications of Wireless Sensor Network*. LIP6//CNRS Université Pierre et Marie Curie.
- [12] M. N.ABDEDDAIM. *Analyse des performances d'un réseau de capteurs exploitant le standard IEEE 802.15.4*. Université de Grenoble, 2012.

- [13] G. SOUSA. *Etude en vue de la réalisation de logiciels bas niveau dédiés aux réseaux de capteurs sans fil : microsystème defichiers*. Université Blaise Pascal Clermont II, 2008.
- [14] P. MUHLETHALER. *802.11 et les réseaux sans fil*. Eyrolles, 2002.
- [15] C.S. RAGHAVENDRA B. KRISHNAMACHARI. *Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks*. In *Performance, Computing, and Communications, 2004 IEEE International Conference*.
- [16] <http://www.ieee802.org/15/pub/TG4.html> IEEE 802.15.4-2006 standard.
- [17] D. DHOUTAUT. *Etude du standard IEEE802.11 dans le cadre des réseaux ad hoc : de la simulation à l'expérimentation*. Thèse de Doctorat, Institut National des Sciences Appliquées de Lyon, 2003.
- [18] B. GRONVALL A. DUNKELS and T. VOIGT. *Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors*. Swedish Institute of Computer Science (SICS), 2004.
- [19] A. DUNKELS. *The ContikiMAC Radio Duty Cycling Protocol*, Swedish Institute of Computer Sciences. 2011.
- [20] <http://www.Contiki-os.org/start.html>. *get started with contiki*. 2013.

## Résumé

Les réseaux de capteurs sans fil (RCSFs) sont une nouvelle technologie, qui a surgis après les grand progrès technologiques concernant le développement des nœuds capteurs, des processeurs puissants et des protocoles de communication sans fil, les domaines d'application des RCSFs nécessitent une forte contrainte temporelle que doit assurer par le réseau, ainsi la bonne exploitation des capteurs sur le plan énergétique. Ce mémoire porte sur l'adaptation du protocole CSMA/CA au temps réel. Tel que implémenté dans le système d'exploitation Contiki 2.7, le protocole CSMA/CA semble non adapté au temps réel, à cause des retransmissions intemporelles et la consommation d'énergie qui est engendrée. Notre objectif est de remédier ces problèmes en proposant deux solutions.

**Mots clés :** Contiki, cooja, csma/ca, couche MAC, norme IEEE802.15.4, RCSFs, consommation d'énergie.

## Abstract

The Wireless Sensors Network (WSNs) are a new technology, which emerged after great technological progress concerning the development of the nodes sensors, the powerful processors and the communications protocols without wire, the applicability of WSNs require a strong temporal constraint which must ensure by the network, thus good exploitation of the sensors on the energy level. This memory relates to the adaptation of protocol CSMA/CA to the real time. As implemented in the operating system Contiki 2.7, protocol CSMA/CA seems not adapted to the real time, because of the timeless retransmissions and the consumption of energy which is generated. Our objective is to cure these problems by proposing two solutions.

**Keywords :** Contiki,cooja, csma/ca, MAC layer, standardIEEE 802.15.4, WSNs, consumption of energy.