Mécanismes auto-adaptatifs pour la gestion de la Qualité de Service dans les réseaux de capteurs sans fil

Bilel Nefzi

Nancy-Université INPL
Institut National Polytechnique de Lorraine

ÉCOLE DOCTORALE IAEM

Département de formation doctorale en informatique

Mécanismes auto-adaptatifs pour la gestion de la Qualité de Service dans les réseaux de capteurs sans fil

T H È S E

présentée et soutenue publiquement le 21 septembre 2011

pour l'obtention du

Doctorat de l'Institut National Polytechnique de Lorraine (spécialité informatique)

par

Bilel NEFZI

Thèse dirigée par Ye-Qiong SONG et

préparée au LORIA, Projet TRIO

Jury :

Rapporteurs :	André-Luc Beylot	-	Professeur à l'IRIT-INPT
	Pascale Minet	-	CR1-HDR à l'INRIA Rocquencourt
Autres membres :	Hervé Guyennet	-	Professeur à l'UFC à Besançon
	Ye-Qiong SONG	-	Professeur à l'INPL
	Sylvain petitjean	-	Directeur de recherche à l'INRIA EST

Loria Laboratoire lorrain de recherche en informatique et ses applications

Laboratoire Lorrain de Recherche en Informatique et ses applications – UMR7503

Remerciements

Je tiens à exprimer ma reconnaissance et ma gratitude à toutes les personnes qui sans lesquelles ce travail de thèse n'aurait pas pu voir le jour.

Je voudrais tout d'abord remercier tous les membres du jury pour la grande attention qu'ils ont bien voulu porter à mon travail.

Je tiens à exprimer mes sincères remerciements et ma reconnaissance à Mr. Ye-Qiong Song d'avoir dirigé ce projet de thèse dans un cadre communicatif et amical. Ses précieux conseils, son écoute et sa patience envers mes questions m'ont été profitable pour la réalisation de ce travail.

Je remercie Mme. Fraçoise Simonot, professeur à l'école des mines de Nancy et chef du laboratoire LORIA, pour m'avoir accueillie dans son équipe de recherche TRIO et pour l'esprit formidable qu'elle a su instauré au sein de l'équipe.

Mes plus respectueux remerciements vont à mes rapporteurs Madame Pascale Minet, CR INRIA et habilitée à diriger des recherches, et Monsieur André-Luc Beylot professeur à l'ENSEEIHT et chef de l'équipe ERIT, d'avoir accepté de lire et d'évaluer mon travail de thèse, pour tout le temps qu'ils m'ont consacré et pour leur remarques qui m'ont aidé à améliorer la qualité de ce manuscrit.

Je remercie également les membres du jury Monsieur Hervé Guyennet, professeur à l'université de Franche-Comté, et Monsieur Sylvain Petitjean, DR INRIA, pour le temps qu'ils ont bien voulu consacrer à l'évaluation de cette thèse.

Mes amicaux remerciements vont à tous les membres de l'équipe TRIO notamment professeur René Schott, Liliana CUCU-GROSJEAN, Nicolas Navet, Lionel Havet, Maha Idrissi Aouad, Najet Boughanmi, Adrien Guenard, Luca Santinelli, Hugo Cruz Sanchez, Dawood Khan, Dorin Maxim, Christian Maxim, Badr El Ahmad, Patrick Meumeu Yomsi et Kodé Lo pour leurs amitiés et leurs apports scientifiques à mes travaux.

Enfin, je remercie toute ma famille à qui je dédie ce travail. Je remercie Anna El Tahchy pour tout. Merci à tous les amis, en particulier Walid Fdhila, Aymen Baoueb, Dawood Khan, Karim Dahmen, Oussama Dabbebi, Najet Boughanmi, Mohamed Kort, Aurore Ko, Bou Baker Nenni, Issam Ben Salem, Wahiba Tawali, Mehdi El Félhi, Mohamed Tlig et Ines Noussa.

A mon père Habib, ma mère Chedlia, mes deux sœurs Manel et Sabrine, mon petit frère Atef et ma chère Anna

Table des matières

Acronymes

API Application Programming Interface

BP Backoff Period

CAP Contention Access Period

CCA Clear Channel Assessment

CDMA Code Division Multiple Access

CoSenS Collect the Send burst Scheme

CFP Contention Free Period

CSMA/CA Carrier Sense Multiple Access/Collision Avoidance

CW Contention Window

DD Diffusion Dirigée

EDF Earliest Deadline First

FFD Full Function Device

GPS Global Positioning System

GTS Guaranteed Time Slots

LPL Low Power Listening

PAN Personal Area Network

PAPS Premier Arrivé Premier Servi

PF Priorité Fixe

QdS Qualité de Service

RA Routage en Arbre

R2A Routage en Arbre Amélioré

RAM Routage en Arbre Modifié

RFD Reduced Function Device

RTS/CTS Request To Send/Clear To Send

SCSP Sleep, Collect then Send Protocol

SP Sleep Period

TCP/IP Transmission Control Protocol/Internet Protocol

TER Taux d'Échéances Respectés

TDMA Time Division Multiple Access

TP Transmission Period

TST Taux de Succès de Transmissions

WP Waiting Period

Chapitre 1

Introduction et problématique

Sommaire

1.1 Réseau de capteurs sans fil

1.1.1 Définition

Les capteurs sont devenus des éléments incontournables dans tous les systèmes où les informations issues de l'environnement extérieur sont nécessaires pour évaluer et agir.

De ce fait, avoir une connaissance précise et complète sur ce milieu exige le déploiement de plusieurs capteurs, et si possible, la conjugaison de toutes les informations renvoyées pour optimiser l'action à entreprendre.

Il n'y a pas si longtemps, la seule solution pour acheminer les données du capteur jusqu'au contrôleur chargé de la surveillance était le câblage qui avait comme principal défaut d'être coûteux et encombrant. Aujourd'hui, le savoir acquis dans les technologies des Microsystèmes électromécaniques (MEMS), des communications sans fil et de l'électronique numérique a permis la conception et le développement de nœuds capteurs de petites taille, communiquant d'une manière sans fil et ayant un faible coût de fabrication et une faible consommation d'énergie [Akyildiz 2010].

Un nœud capteur est constitué d'une source d'alimentation (souvent une batterie), d'un ou plusieurs capteurs, d'une unité de traitement et d'un système de communication sans fil. Un réseau de capteurs sans fil est un ensemble de nœuds capteurs,

appelées capteurs ci-après, communiquant entre eux. Le but du déploiement d'un tel réseau est la collecte de données de mesure d'une zone géographique, et dans certains cas l'action sur celle-ci (on parle alors de réseaux de capteurs-actionneurs).

Les applications des réseaux de capteurs sans fil sont nombreuses et variées. Grâce à leur facilité de déploiement et leur faible coût, ces réseaux sont en train de pénétrer tous les aspects de notre vie.

1.1.2 Domaines d'application

Akyldiz et al. [Akyildiz 2010] proposent cinq grandes familles d'applications. D'abord, on a les applications militaires. Leurs objectifs incluent la surveillance des forces amies, des équipements, des munitions et du champs de bataille, la détection des attaques (biologiques, chimiques, *etc.*), la reconnaissance des forces ennemies, l'évaluation des dommages, *etc.* Un exemple d'application militaire est VigilNet [He 2006]. C'est un réseau de surveillance à grande échelle conçu pour le pistage de cibles dans des environnements difficiles. Ensuite, on a les applications pour l'environnement comme par exemple la détection de feux dans les forêts, la détection d'inondation, l'agriculture de précision, le recensement et la surveillance d'animaux, *etc.* Le projet "Great Duck Island" [Mainwaring 2002] est un exemple d'application pour l'environnement. C'est le résultat d'une collaboration entre "College of Atlantic" et "Intel Research Laboratory" à Berkeley pour étudier la distribution et l'affluence des oiseaux de mer à Great Duck Island, Maine. Après, on a les applications médicales comme celles permettant la télé-surveillance des patients et de leurs fonctions vitales, le repérage des patients, des médecins et du matériel dans un hôpital, *etc.* Le projet "Appartement intelligent pour l'assistance à la personne" [inf 2010] est un exemple d'application médicale. Ce projet, qui est en cours de développement, est mené par les équipes Trio, Madynes, Maia et Cortex du Loria et vise des applications relatives à la domotique et à l'assistance à la personne dans des cadres aussi bien professionnels que médicaux ou privés (exemple : l'aide cognitive pour des professions où la surcharge en information est grande, le maintien à domicile des personnes âgées ou dépendantes). Ce projet fait partie aussi de la quatrième famille, celle des applications pour la maison. Enfin, on a les applications industrielles. La supervision, la localisation et l'inventaire des équipements industriels sont des exemples d'applications industrielles. Par exemple, La société fireflies-rtls [Fireflies-rtls 2010] propose une solution de localisation de bien en temps réel destinée au industriels.

Selon le type d'application et son environnement de déploiement, les contraintes sur le trafic généré sont différentes. Les deux paragraphes suivants détaillent ces deux points.

1.1.3 Types des flux de trafic de données

Les modèles des flux de trafic de données les plus utilisés dans les réseaux de capteurs sans fil sont en nombre de quatre. Ils sont décrit ci-après.

- **Périodique** : chaque capteur dans le réseau envoie périodiquement des paquets aux destinations. La plupart des applications génèrent du trafic périodique.
- **Événementiel** : les capteurs envoient des paquets aux destinations lorsqu'ils détectent un événement. Par exemple, dans les applications de surveillance, un capteur n'envoie des paquets que s'il détecte une présence dans la zone qu'il surveille.
- **Requête/réponses** : un contrôleur du réseau envoie une requête à un sous-ensemble du réseau pour demander une information particulière. Cet ensemble envoie des paquets aux destinations en réponse à cette requête. Par exemple, dans une application de télé-surveillance médicale, le médecin envoie une requête pour demander l'état actuel des signes vitaux d'un patient.
- **Hybride** : dans lequel une application utilise un ou plusieurs types de trafic, qu'on vient de décrire, à la fois. De plus en plus d'applications utilisent ce modèle de nos jours. Par exemple, l'application de télé-surveillance médicale peut utiliser tous les types de trafic à la fois. Les capteurs envoient d'une façon périodique l'état des fonctions vitales du patient. Si entre temps un problème est détecté, un événement est transmis au professionnel de santé. Ce dernier peut envoyer une requête pour vérifier l'état du patient à un instant donné.

Nous avons remarqué à la fin de la sous-section précédente que la nature de l'application implique des contraintes additionnelles liées au trafic. Dans l'application de télé-surveillance médicale par exemple, le trafic événementiel signale un problème dans un ou plusieurs signes vitaux d'un patient. Ainsi, nous pouvons qualifier ce trafic comme critique par rapport au trafic périodique qui met juste à jour les données vitales des patients. Ces contraintes définissent le niveau de service exigé par une application. C'est dans ce cadre que la notion de Qualité de service (QdS) prend tout son sens.

1.1.4 Contraintes

Bien qu'ils fassent partie du domaine des réseaux ad-hoc, les réseaux de capteurs sans fil forment une branche de recherche à part. En effet, les contraintes liées à ce type de réseaux sont différentes de celles du domaine classique, ce qui explique le fait que tous les protocoles et les algorithmes déjà développés pour les réseaux sans fil classiques ne soient pas adaptés aux réseaux de capteurs sans fil. Cependant, le média sans fil reste une contrainte commune. Il diffuse par nature, n'est pas fiable et son état varie dans le temps. On présente ci-après les autres contraintes liées aux réseaux de capteurs sans fil.

Le premier facteur est celui de l'échelle. Le nombre de capteurs déployés pour étudier un phénomène est généralement de l'ordre de centaines ou de milliers. La densité des nœuds est aussi élevée. Ainsi les algorithmes développés doivent avoir une très bonne capacité de passage à l'échelle : c'est la première différence avec les réseaux sans fils classiques où le nombre de stations ne dépasse généralement pas une centaine.

La deuxième contrainte est liée au coût de production. Puisque les capteurs sont déployés par milliers, leur coût de production doit rester aussi faible que possible. Ceci induit d'autres contraintes, principalement matérielles (faible capacité de mémoire, faible puissance de calcul, *etc.*) puisque optimiser le coût de production revient à optimiser le coût des composants matériels.

La troisième contrainte qui est, sans doute, la plus importante est celle liée à la consommation d'énergie. La source principale d'énergie d'un nœud est, dans la plupart des cas, une batterie à durée de vie limitée et souvent irremplaçable. C'est la partie communication qui consomme le plus d'énergie. Ainsi les protocoles et les algorithmes doivent prendre en considération le facteur énergie souvent négligé dans les protocoles développés pour les réseaux sans fil classiques (dans ces réseaux l'énergie est supposée infinie).

La quatrième contrainte est liée aux deux précédentes. Le faible coût et l'utilisation d'une batterie rendent le capteur sujet aux pannes beaucoup plus fréquemment. Ces pannes peuvent survenir à cause d'un manque d'énergie, de l'interférence de l'environnement extérieur, de problèmes logiciels, *etc.* La panne peut être temporaire ou définitive. Dans tous les cas, elle ne doit pas affecter le fonctionnement normal du réseau.

Finalement, s'ajoute à toutes ses contraintes déjà fortement contraignantes le besoin de garantir un niveau de service suffisant pour les applications imposant des contraintes à leurs flux de données.

1.2 La qualité de service

1.2.1 Définition

La recommandation E.800 (09/2008) de l'union internationale des télécommunications (UIT) a définit la QdS comme étant l'"*Ensemble des caractéristiques d'un service de télécommunication qui lui permettent de satisfaire aux besoins explicites et aux besoins implicites de l'utilisateur du service*". Une caractéristique est définie comme étant la "*propriété qui aide à faire la distinction entre les individus d'une population donnée*". Elle peut être quantitative ou qualitative. La QdS demandée par un utilisateur reflète ses besoins en matière de qualité de fonctionnement eux même exprimés en termes descriptifs (critères) et énumérés par ordre de priorité, une valeur préférée étant donnée pour chaque critère (E.800 (09/2008)). L'administrateur du réseau quantifie cette demande en utilisant des métriques de QdS tel que la bande passante, le délai, la gigue, la fiabilité, *etc.*

1.2.2 Mécanismes de QdS

Les mécanismes traditionnels de QdS les plus importants sont les politiques de gestion de files d'attente, le lissage du trafic, le contrôle du trafic, le contrôle d'admission et le contrôle de congestion [Ferguson 1998]. Ils sont décrit brièvement ci-après.

Politiques de gestion de files d'attente
Les files d'attente font partie intégrante des réseaux informatiques car ces derniers opèrent selon le paradigme stocker puis transmettre, impliquant le stockage de paquets en attendant leur service. Par conséquent, les disciplines de files d'attente ont joué, et continuent à jouer, un rôle important dans la différenciation de service [1]. WFQ [Parekh 1993] et l'ordonnancement par priorité [Liu 1973] sont des exemples de disciplines de files d'attente. La différenciation de service à l'aide des politiques de gestion de files d'attentes est moins efficace dans le cas des réseaux sans-fil à cause des spécificités du média sans-fil. C'est pour cela que d'autres moyens de différenciation de service ont vu le jour dans ces réseaux.

Lissage du trafic
Le lissage du trafic est la pratique permettant de contrôler le volume du trafic entrant dans le réseau, ainsi que le contrôle de la vitesse à laquelle il est transmis. Il est nécessaire d'identifier les flux de trafic au point d'entrée du réseau afin d'offrir un mécanisme de lissage différent selon leurs caractéristiques. Les deux mécanismes les plus connus sont le sceau percé et le seau à jetons.

Contrôle du trafic
Le contrôle du trafic est la pratique permettant la vérification, au niveau de chaque saut du réseau, de la conformité d'un trafic circulant au paramètres pré-négociés à l'entrée du réseau et la prise des actions requises en cas de problème tel que la suppression de paquets ou la dégradation de la QdS.

Contrôle d'admission
Le contrôle d'admission, au sens le plus primitif, est la simple pratique permettant de décider si un trafic est admis dans le réseau ou pas. Par exemple, ceux qui ont payé peuvent envoyer du trafic, ou un trafic donné peut être admis selon les conditions du réseau, ou encore un trafic ne peut être admis que si le réseau est capable de fournir la QdS demandée par ce trafic. Le contrôle d'admission est une partie importante de chaque implémentation de la QdS, car si on n'a aucun contrôle sur le trafic entrant dans le réseau, la congestion est inévitable et le réseau sera incapable de fournir le service demandé par tous ses utilisateurs.

1. La différenciation de service, c'est fournir un traitement différent selon les caractéristiques du service demandé.

Contrôle de congestion
Le lissage du trafic, le contrôle du trafic et le contrôle d'admission font partie de l'approche préventive pour le contrôle du trafic. Elle consiste à prendre des mesures a priori pour minimiser les chances d'apparition d'une congestion. Le contrôle de congestion fait partie de l'approche réactive pour le contrôle du trafic. Dans cette approche, il s'agit d'accepter le maximum de trafic possible tant qu'il n'y a pas de congestion. Dès qu'une congestion se déclare, le réseau est alors autorisé à diminuer le débit d'une connexion ou à supprimer des paquets. Le mécanisme de fenêtre de congestion du protocole TCP/IP [Jacobson 1988] est un exemple bien connu du contrôle réactif.

1.2.3 Problématique de la QdS dans les réseaux de capteurs sans fil

Implémenter un ou plusieurs mécanismes de QdS sur un réseau sans fil, dense, hautement dynamique, composé de nœuds ayant une faible capacité de mémoire et processeur, à forte contrainte énergétique et utilisant des modèles de trafic non traditionnels est une tâche difficile. Néanmoins elle est nécessaire car plusieurs types de trafic peuvent circuler dans le réseau dont certains ont une forte contrainte (de délai par exemple). La relation énergie/QdS est particulièrement intéressante, car les deux objectifs sont antagonistes ; le premier vise la minimisation de la consommation d'énergie et l'autre vise la satisfaction des contraintes liées au trafic.

La différenciation de service est la principale technique utilisée dans les réseaux de capteurs sans fil pour le support de la QdS. Cependant, utilisée seule, elle n'est pas suffisante pour obtenir des garanties déterministes. La QdS dans les réseaux de capteurs sans fil est donc largement best-effort. Dans le reste de la section nous présentons les mécanismes de différenciation de service.

différenciation de service

Le premier volet dans la différenciation de service est la classification du trafic, *i.e.* la spécification de la priorité de chaque paquet circulant dans le réseau. Le choix peut être statique ou dynamique. Dans le premier cas, une priorité est attribuée au paquet dès sa création. Elle ne change pas au cours de sa traversée du réseau. L'attribution peut être en fonction du type de trafic (un événement est plus prioritaire qu'un message périodique), de la donnée captée (un message contenant la mesure de la température d'un patient est plus importante qu'un message de localisation d'un appareil médical) ou n'importe quel ordre spécifié par les concepteurs du réseau. Dans le deuxième cas, la priorité varie pendant la traversée du paquet. Elle dépend par exemple, du nombre de sauts traversés, du nombre de sauts restants, de l'énergie restante d'un nœud, de l'échéance du paquet, de la charge subie par un nœud ou de n'importe quelle combinaison de ces critères.

Le deuxième volet concerne les mécanismes mis en œuvre pour la différenciation

de service. Dans les réseaux de capteurs sans fil, elle se base essentiellement sur la variation des paramètres du protocole d'accès en fonction de la priorité du trafic (voir 2.1.6).

QdS garantie, probabiliste ou best-effort ?

Fournir de la QdS garantie dans les réseaux de capteurs sans fil relève de l'utopie. D'une part, les réseaux de capteurs sans fil sont hautement dynamiques et denses. Ainsi, pour offrir des garanties absolues en terme de délai ou de fiabilité, il est nécessaire d'utiliser des protocoles d'accès déterministes et synchronisés, des protocoles de routages sophistiqués impliquant beaucoup d'échanges de données et des mécanismes de lissage de trafic, de contrôle du trafic et de contrôle d'admission. D'autre part, les contraintes d'énergie, de mémoire et de ressources processeur rendent impossible la mise en œuvre de tous ces protocoles. Une première approche possible consiste à fournir de la QdS best-effort. Cela consiste à fournir un traitement différent selon les caractéristiques du service demandé sans pour autant garantir des performances déterministes. Plus besoin d'avoir un protocole déterministe ni d'approche préventive pour le contrôle du trafic. Une deuxième approche consiste à fournir une QdS probabiliste, qui offre un traitement différent selon les caractéristiques du service demandé tout en garantissant d'une manière probabiliste une ou plusieurs métriques de performance. Typiquement, il s'agit de garantir qu'une métrique m, délai par exemple, ne dépasse pas la valeur v, 0.5 s par exemple, dans $P\%$ des cas, 60% par exemple. La QdS probabiliste constitue une incrémentation de celle best-effort.

L'idée que nous soutenons consiste à développer, dans un premier temps, une suite de mécanismes pour la gestion de la QdS best-effort qui soit simple, peu coûteuse et suffisamment flexible pour pouvoir l'étendre aisément.

1.3 Problématique et contributions de la thèse

L'objectif de cette thèse est de fournir un ensemble de solutions permettant le support de la QdS dans les réseaux de capteurs sans fil. Comme mentionné à la fin de la section précédente, l'implémentation de cet ensemble doit être *simple*, *pas coûteuse*, *flexible* et ne nécessitant *pas d'effort de configuration*. Cependant, une première analyse du comportement d'un réseau de capteurs sans fil, nous a permis de remarquer qu'en l'absence d'un événement, le réseau fonctionne normalement, *i.e.* il n'y a ni surcharge ni contraintes fortes sur le trafic qui circule. Lorsqu'un événement survient, tous les nœuds s'activent pour informer le(s) destinataire(s) le plus rapidement possible et avec plus de précision (ce constat est détaillé dans la section 5.1 du chapitre 5). On imagine que dans le premier cas, le réseau se soucie de préserver l'énergie et que dans le deuxième cas, il privilégie le délai et la fiabilité de transmission. Ainsi, les mécanismes doivent aussi s'*auto-adapter aux évolutions du réseau*.

Les contributions de la thèse sont :

- Une analyse des principales techniques MAC et routage. Cette étude montre les avantages et les inconvénients de ces protocoles, ce qui nous a permis de développer de nouvelles techniques répondant mieux aux besoins des applications.
- Le développement d'un nouveau protocole MAC, appelé CoSenS [2]. Son objectif est d'améliorer les performances du protocole d'accès CSMA/CA et de faciliter l'implémentation de mécanismes de QdS. CoSenS, constitue le pilier de base pour le support de la QdS.
- Priority-CoSenS, P-CoSenS, une première extension de CoSenS présentant une technique de différenciation de service basée sur les disciplines des files d'attente. Notre approche diffère de l'approche standard qui se base sur la variation des paramètres du protocole d'accès pour la différenciation de service.
- Sleep-CoSenS, S-CoSenS, un ensemble de protocoles qui traite le compromis entre la minimisation de la consommation d'énergie et la préservation d'un débit suffisant pour l'application fonctionnant sur le réseau. S-CoSenS utilise des optimisations inter-couches pour améliorer la fiabilité de la transmission des données.

Chaque contribution correspond à un chapitre. Le dernier chapitre conclut la thèse.

1.4 Structure de la thèse

La suite de la thèse est organisée comme suit. Le chapitre 2 présente l'état de l'art sur les solutions MAC et routage proposées pour les réseaux de capteurs sans fil. Le chapitre 3 présente CoSenS ainsi qu'une étude détaillé de ses performances par simulation et par implémentation sur une plate-forme matérielle. Le chapitre 4 détaille une première extension à CoSenS qui consiste à l'implémentation d'un mécanisme de différenciation de service. Dans le chapitre 5, on présente une deuxième extension de CoSenS qui traite le compromis energie/débit. Le dernier chapitre conclut cette thèse et présente les perspectives.

2. CoSenS, c'est un acronyme pour "Collect then Send burst Scheme"

Chapitre 2

État de l'art

Sommaire

Les réseaux de capteurs sans fil présentent des caractéristiques et des contraintes différentes des réseaux sans fil classiques. Cela a conduit au développement de nombreux protocoles MAC et routage qui tiennent en compte de ces caractéristiques.

Ce chapitre présente les principaux protocoles MAC et routage développés pour les réseaux de capteurs sans fil. Il n'a pas vocation de résumer tous les travaux existants. Il présente plutôt les principales techniques utilisées pour répondre aux exigences des réseaux de capteurs à travers quelques exemples de protocoles développés.

Le chapitre est organisé comme suit. La première section propose une étude de l'état de l'art des protocoles MAC. La deuxième section présente les travaux réalisés sur le routage. La troisième section conclut le chapitre.

2.1 Protocoles MAC

Les protocoles MAC peuvent être classés en trois catégories ; basés sur la contention, ordonnancés ou hybrides. Les trois sous-sections suivantes décrivent ces trois catégories à travers quelques exemples de protocoles MAC. La sous-section d'après détaille

des exemples de protocoles MAC proposant une gestion de la QdS. La sous-section suivante présente les efforts de standardisation de la couche MAC. Dans la dernière sous-section, nous proposons une analyse de l'état de l'art des protocoles MAC.

2.1.1 Protocoles basés sur la contention

2.1.1.1 S-MAC

S-MAC [Ye 2002] est un protocole basé sur la contention. Un nœud utilisant S-MAC enchaîne un cycle divisé en une partie active et une partie inactive. Durant la partie active, il écoute le canal et reçoit et envoie des données. Durant la partie inactive il désactive son récepteur pour préserver l'énergie. Les nœuds se synchronisent entre eux pour qu'ils se réveillent et se rendorment en même temps. Ainsi ils adoptent un rendez-vous commun. Le fonctionnement de cet algorithme est comme suit :

- Un nœud qui démarre écoute le canal.
  - ◇ S'il découvre un nœud qui a déjà choisi ses périodes actives et inactives, il adopte le même choix.
  - ◇ Sinon, il choisit lui-même ses périodes et annonce ses choix périodiquement.
- Si un nœud découvre qu'il n'a pas les mêmes périodes que son voisin, il adopte les deux.

La figure 2.1 montre un exemple de fonctionnement de S-MAC. Les nœuds A et C ne s'entendent pas. Le nœud B entend les deux autres. Le nœud A démarre le premier, suivi du nœud C. Ainsi, chacun va choisir un ordonnancement différent. Lors du démarrage du nœud B, il choisit de suivre le nœud A. Il se rend compte ensuite que le nœud C a adopté un ordonnancement différent. Il décide alors de suivre les deux.

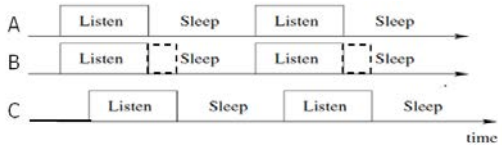


FIGURE 2.1 – Algorithme de rendez-vous commun dans S-MAC.

S-MAC nécessite donc une synchronisation entre les nœuds pour assurer son bon fonctionnement. Cependant, elle n'est pas aussi exigeante que pour un protocole du type TDMA. En effet, S-MAC utilise des périodes actives assez longues, de l'ordre de centaines de millisecondes, par rapport à la dérive de l'horloge interne. La partie active est elle-même divisée en deux parties ; une partie pour la transmission et la réception des messages de synchronisation et une autre pour la transmission et la réception des paquets de données (Fig. 2.2). La figure montre aussi les différentes possibilités de transmission. En l'occurrence, "Sender 1" n'a pas de données à transmettre mais envoie un message de synchronisation (SYNC) et se rendort après. "Sender 2" a une donnée à transmettre et envoie donc le message RTS. "Sender 3" envoie un message de synchronisation et une donnée. Notons que l'envoi d'un message de synchronisation ne se fait pas systématiquement dans chaque période active,

la décision est aléatoire.

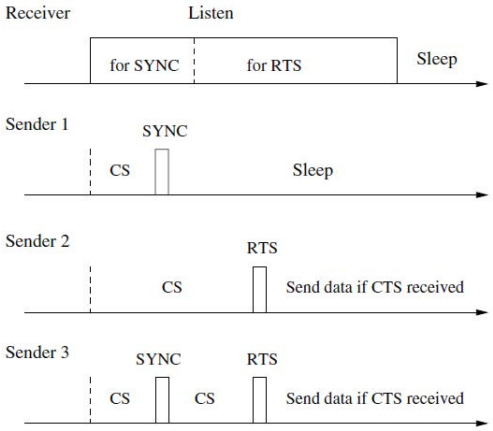


FIGURE 2.2 – Illustration de la partie active dans S-MAC ainsi que les différentes possibilités de transmission. CS désigne la détection de la porteuse [Ye 2002].

Pour la transmission de données, S-MAC utilise deux mécanismes ; RTS/CTS et la détection virtuelle de la porteuse (DVP) [1], les deux inspirés de IEEE 802.11 [IEEE 2007]. La différence par rapport à IEEE 802.11 est que si un nœud écoute un paquet RTS ou CTS, il désactive sa radio pendant la durée de transmission mentionnée. Ceci permet d'éviter l'écoute du canal pendant toute la durée de transmission et ainsi améliorer sa consommation d'énergie au prix d'une DVP moins performante (si la transmission d'un paquet échoue, la durée annoncée dans le paquet RTS ne sera pas respectée). Pour transmettre un paquet long, S-MAC le fragmente en plusieurs paquets et les envoie en un seul burst. Il utilise un seul RTS dans lequel il mentionne la durée totale de transmission de tous les fragments.

2.1.1.2 T-MAC

T-MAC [van Dam 2003] est une amélioration de S-MAC. Le protocole propose une gestion plus dynamique de la partie active. Un nœud transmet tous les messages en un burst au début de la partie active. De plus, l'intervalle maximal de contention est fixé (CW dans IEEE 802.11). Ainsi toutes les transmissions sont décalées au début de la partie active. De cette façon, un nœud se rendort s'il ne reçoit rien pendant un temps TA. La figure 2.3 illustre ce comportement pour un nœud donné. Le trafic "normal" indique les instants de génération des messages par la couche application. Une flèche montante indique une génération par le nœud en question. Celle descendante indique un message généré par un autre nœud à destination du nœud en question. La figure montre bien que les messages générés ne seront transmis ou reçus que pendant la période active. En plus, dès que le nœud reste inactif pendant TA, il se rendort. Ainsi les messages suivants attendront la prochaine période active.

1. Virtual Carrier Sens en anglais

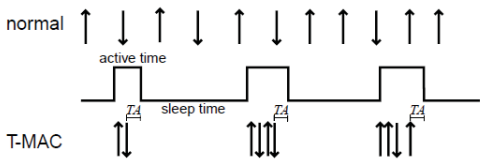


FIGURE 2.3 – Exemple d'opération de T-MAC [van Dam 2003].

La notion de burst dans T-MAC est sensiblement identique à celle de S-MAC ; un seul RTS/CTS pour l'ensemble des messages à transmettre à un seul nœud. Ceci veut dire que pour transmettre un ensemble de messages à un nœud, il suffit de gagner l'accès au canal une seule fois.

2.1.1.3 B-MAC

B-MAC [Polastre 2004] est un dérivé du protocole CSMA. Il gère la consommation d'énergie en utilisant une technique appelée "écoute à faible puissance" (LPL [2]). Chaque nœud échantillonne le temps en intervalles réguliers. Le nœud est par défaut en mode inactif. Au début de chaque intervalle il s'active, allume sa radio et vérifie l'état du canal (CCA). S'il détecte une activité, il reste actif pendant le temps nécessaire pour recevoir le paquet. Sinon, il se rendort. Après la réception, le nœud repasse en mode inactif. Si un paquet n'est pas reçu, un minuteur force le nœud à se rendormir. Du coté de l'émetteur, chaque transmission d'un paquet est précédée par la transmission d'un préambule. La taille du préambule doit correspondre au moins à l'intervalle d'échantillonnage. De cette façon le récepteur est averti et reçoit le paquet de données. Un exemple de fonctionnement de B-MAC est illustré par la figure 2.4. La couche MAC du nœud B reçoit un message à destination de A ; l'étoile indique l'instant de réception du paquet par la couche MAC. L'ordonnancement des réveils et des endormissements de A est donné par la première ligne de la figure. B se réveille et choisit une période de backoff, BP. A la fin de cette période il teste si le canal est encore libre. Si c'est le cas il transmet le préambule suivi par le packet. A détecte le préambule et reste actif. Il reçoit ainsi le paquet P.

La procédure de vérification de l'état du canal doit être précise pour obtenir de bonnes performances en utilisant le LPL. Pour cela, B-MAC emploie une technique d'estimation du bruit du canal pour améliorer les décisions rendues par cette procédure.

Les auteurs proposent aussi une analyse de B-MAC permettant de dimensionner les paramètres du protocole, comme la taille du préambule et la durée de l'intervalle, avant le déploiement en fonction de la charge estimée et du nombre de nœuds voisins pour augmenter la durée de vie du réseau. De plus, l'implémentation de B-MAC fournit quelques interfaces pour les couches supérieures leur permettant de modifier ces mêmes paramètres. Notons que la modification de ces paramètres au cours du

2. abréviation du terme anglais "Low Power Listening"

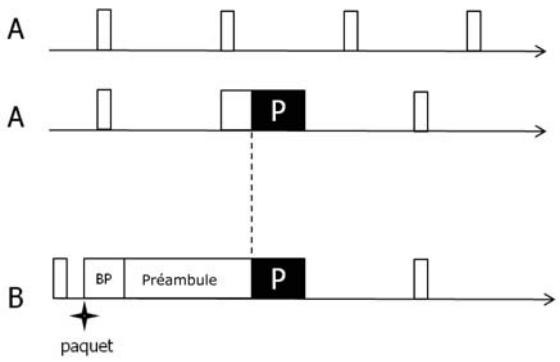


FIGURE 2.4 – Exemple de transmission d'un paquet dans B-MAC.

fonctionnement du réseau nécessite une reconfiguration globale de tous les nœuds.

2.1.1.4 WiseMac

WiseMac [El-Hoiydi 2004] est basé sur CSMA non persistant et utilise une technique appelée "échantillonnage de préambule" pour minimiser la consommation d'énergie causée par l'écoute du canal ("idle" listening). La technique d'échantillonnage de préambule est sensiblement identique à celle de l'écoute à faible puissance utilisée par B-MAC. Les deux techniques ont été développées relativement au même moment. WiseMAC propose une amélioration de cette technique qui consiste à apprendre les temps de réveil des nœuds voisins. Ceci permet d'utiliser un préambule de plus courte durée, transmise juste avant le début de l'écoute du canal du nœud récepteur. Cette optimisation améliore l'énergie consommée pour la transmission d'un paquet ainsi que le débit maximal du réseau. Les informations de synchronisation sont envoyées dans les acquittements, WiseMAC suppose que toutes les transmissions sont acquittées.

Le mécanisme de synchronisation introduit un risque de collision systématique, dans le cas où plusieurs nœuds envoient régulièrement des paquets au même récepteur. Pour pallier ce problème, les auteurs proposent un mécanisme de réservation basé sur la longueur du préambule. Les nœuds choisissent une taille variable du préambule à transmettre avant le réveil du récepteur. Ainsi, le nœud ayant le plus long préambule gagnera l'accès au support et transmettra ses données le premier.

WiseMAC utilise aussi un bit "encore" dans ces paquets afin d'informer le récepteur s'il y a encore des paquets à lui transmettre par le même émetteur. Ceci permet d'éviter que le récepteur ne se rendorme dès la réception du premier paquet et d'envoyer ainsi un burst de données à un même récepteur en utilisant un seul préambule.

2.1.1.5 X-MAC

X-MAC [Buettner 2006] est une amélioration de la technique d'écoute à faible puissance (LPL). Au lieu d'envoyer le préambule tout entier, X-MAC la divise en un ensemble de petits paquets préambules et les transmet un par un tout en insérant un intervalle de temps entre eux. Chaque paquet préambule contient l'adresse du destinataire du paquet à transmettre. Ainsi, l'intervalle de temps entre deux paquets préambules permet au nœud destinataire d'envoyer un acquittement lorsqu'il reçoit l'un de ces paquets préambules. Une fois l'acquittement reçu par l'émetteur, il transmet le paquet vers le destinataire. Ce mécanisme est illustré par la Fig. 2.5.

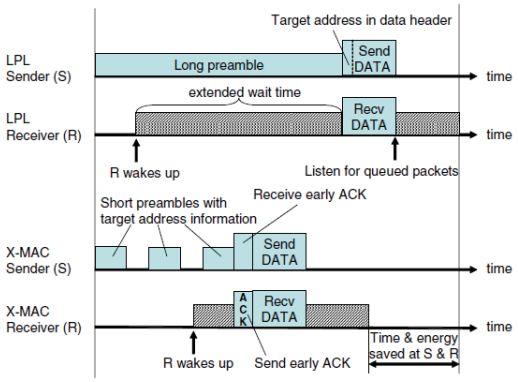


FIGURE 2.5 – Comparaison entre X-MAC et LPL [Buettner 2006].

Cette technique permet de réduire encore plus l'énergie consommée par l'émetteur et les récepteurs. Tout d'abord, l'émetteur n'a pas besoin de transmettre tout le préambule pour que le récepteur se réveille et reste actif. Ensuite, en insérant l'adresse du destinataire, les autres nœuds qui ont été réveillés par le préambule se rendorment aussitôt. Enfin, les dépenses énergétiques de cette technique par rapport à LPL sont moins dépendantes de la densité du réseau.

X-MAC apporte d'autres améliorations par rapport à LPL. Premièrement, il améliore les délais de bout-en-bout. Deuxièmement, il est plus adapté aux "paquetizing radios" comme la Chipcon CC2420. LPL étant développé pour les bits streaming radios comme la Chipcon CC1100 [3]. Troisièmement, il permet d'adapter la durée de sommeil et d'écoute en ligne en fonction du trafic. Pour cela, il utilise une méthode d'estimation du trafic reçu et un tableau déjà pré-calculé fournissant les valeurs optimales de ces deux paramètres en fonction du trafic. Cependant, se basant sur CSMA, les performances de X-MAC se dégradent si le trafic augmente considérablement. La performance en termes de débit et de délai de bout en bout de X-MAC

3. Les "bit streaming radios" récupèrent un flux de bits brut et le transmet directement. C'est le microprocesseur qui se chargera donc de formater le paquet. Dans les "paquetizing radio", au lieu de transmettre un flux de bits brut, la puce radio récupère la trame et rajoute elle même le préambule et le CRC. A la réception elle contrôle elle même le paquet et l'envoie au processeur s'il est bon. Ainsi, l'application ne peut plus contrôler directement les bits envoyés sur le canal.

est bornée par celle de CSMA.

2.1.2 Protocoles ordonnancés

2.1.2.1 PEDAMACS

PEDAMACS [Ergen 2006] est un protocole MAC de type TDMA. Il repose sur deux hypothèses de base. Premièrement, le réseau est composé d'une station de base (SB) et d'un ensemble de capteurs sans fil. La **SB peut atteindre tous les capteurs en un seul saut** alors que les messages générés par les capteurs ont besoin de plusieurs sauts pour atteindre la SB. Deuxièmement, les capteurs génèrent des messages d'une façon périodique.

Le fonctionnement de PEDAMACS s'opère en quatre phases, toutes dictées par la SB. A chaque fois, la SB envoie un message annonçant le temps de début et de fin de chaque phase. La première phase est celle d'*apprentissage de la topologie*, durant laquelle tous les capteurs déterminent leurs topologies locales, à savoir leurs voisins, leur père et les nœuds interférents. Ensuite, la phase de *collection de la topologie* commence. Durant cette phase, tous les nœuds transmettent leurs topologies locales à la BS. Après l'apprentissage de la topologie du réseau (les deux premières phases), la SB ordonnance les slots de transmission et de réception de tous les nœuds du réseau et diffuse le résultat. Cette phase s'appelle la *phase d'ordonnancement*. A l'issue de cette phase tous les nœuds commencent à suivre l'ordonnancement et peuvent transmettre et recevoir des données. Notons que la synchronisation se fait toujours par rapport aux messages transmis par la SB puisqu'elle est capable d'atteindre tout le réseau en un seul saut. Les nœuds peuvent transmettre des données de surveillance ainsi que des données de mise à jour de topologies (détection de nouveaux voisins par exemple). La dernière phase est celle d'*ajustement*, durant laquelle les nœuds et la SB mettent à jour leurs topologies locales et l'ordonnancement des slots, respectivement. Les phases d'apprentissage et de collection de la topologie sont exécutées une seule fois alors que la phase d'ajustement s'exécute d'une manière périodique (chaque 10 min par exemple).

PEDAMACS élimine le problème de synchronisation en utilisant un nœud capable d'atteindre tout le réseau. L'hypothèse adoptée par cette approche en limite son déploiement. L'économie de l'énergie se fait à travers l'utilisation de TDMA par les capteurs sans se soucier de la synchronisation et de l'ordonnancement ; les tâches qui consomment le plus d'énergie dans les protocoles de type TDMA.

2.1.2.2 LMAC et AI-LMAC

LMAC [Hoesel 2004] est un protocole MAC basé sur TDMA. Le temps est divisé en trames, elles même divisées en slots. Chaque nœud se verra assigner un slot pendant lequel il peut transmettre un paquet de données. Pendant son slot, un nœud transmet un message de contrôle suivi éventuellement d'un paquet de donnée

s'il en a un à transmettre. Tous les nœuds doivent se réveiller au début du slot pour recevoir le message de contrôle. Seul le destinataire reste actif pour recevoir le paquet de données par la suite (ils se rendorment tous s'il n'y a rien à transmettre). L'émetteur et le récepteur se rendorment à la fin de la transmission du paquet.

L'ordonnancement des slot se fait d'une façon distribuée. La synchronisation s'effectue pas-à-pas en commençant par le coordinateur du réseau. Les deux opérations se font simultanément au démarrage du réseau comme suit. Le coordinateur démarre le réseau et choisit un slot. Pendant son slot il va transmettre son message de contrôle à tous ses voisins permettant à la fois de les synchroniser et de les informer de l'ensemble des slots libres. Les voisins vont donc se synchroniser puis ils vont choisir à leur tour un slot libre au hasard. Ces nœuds transmettent leurs messages de contrôle pendant la trame suivante, indiquant aussi la liste des slots libres. Si deux voisins choisissent le même slot, les nœuds voisins ont la responsabilité de les informer. L'opération se poursuit après d'une manière incrémentale jusqu'à ce que tous les nœuds soient synchronisés et un slot soit réservé par chacun d'entre eux. Notons qu'un nœud ne gagne le contrôle d'un slot qu'après la consultation des slots libres de tous ses voisins. Le nombre total des slots à réserver est fixé dès le départ.

AI-LMAC [Chatterjea 2004] est une extension de LMAC qui prend en compte les besoins de chaque nœud en terme de bande passante. A l'opposé de LMAC, il autorise un nœud à réserver plusieurs slots. Cependant, la décision de réserver un slot supplémentaire n'est pas prise par le nœud lui même mais par sa hiérarchie en fonction du trafic estimé de chacun de ses fils. Donc finalement c'est le coordinateur du réseau qui décide le premier des slots à allouer à chacun de ses voisins qui eux même distribuent ces ressources à leurs fils. Pour pouvoir utiliser cette technique, chaque nœud maintient une table de statistiques de données réparties en fonction de la nature des données collectées de chaque nœud actif. Ainsi, pour chaque requête générée par les utilisateurs du réseau, chaque nœud, en commençant par la passerelle du réseau, peut prédire la quantité du trafic à recevoir ainsi que sa répartition en fonction de ses fils. Bien sûr, les auteurs supposent que le réseau est organisé d'une manière hiérarchique.

L'algorithme d'ordonnancement et de synchronisation est simple et bien adapté aux réseaux de capteurs. Seulement, les auteurs n'ont pas pris en considération les effets de la propagation des messages de synchronisation. Le protocole considère un nombre fixe de slots à réserver ce qui est le prix à payer pour avoir cette simplicité.

2.1.3 Protocoles hybrides

2.1.3.1 Z-MAC

Z-MAC[Rhee 2008] est un protocole MAC hybride. Il change dynamiquement de mode de transmission entre CSMA et TDMA en fonction de la charge actuelle du réseau. Z-MAC utilise CSMA comme protocole de base pour l'accès au support (en fait il utilise B-MAC) mais emploie un ordonnancement de type TDMA pour

améliorer la résolution de contention entre les nœuds.

Le temps est divisé en slots. Au début du déploiement, chaque nœud se verra attribuer un slot en utilisant DRAND [Rhee 2006], un algorithme distribué d'affectation de slots élaboré par les mêmes auteurs de Z-MAC. DRAND garantit que les slots des nœuds se trouvant à deux sauts entre eux ne se chevauchent pas. Une fois le slot affecté, le nœud l'utilise pendant chaque période préderminée, appelée **trame**. Ce nœud devient le **propriétaire** du slot. A l'opposé de TDMA, un nœud peut envoyer des données à n'importe quel slot avec Z-MAC. Par contre, le nœud propriétaire a la priorité pour envoyer ses données au cours de son slot. La priorité est implémentée en ajustant la fenêtre de contention initiale pour garantir aux nœuds propriétaires un accès privilégié au canal de transmission au cours de leurs slots respectifs.

Bien qu'un nœud propriétaire ait un accès privilégié au cours de son slot, sa priorité n'est pas absolue parce que les transmissions ne démarrent pas forcément au début de chaque slot (à la différence de TDMA, on peut transmettre plusieurs messages dans un slot). Cependant, si un nœud détecte une charge importante, il en informe ces voisins qui à leur tour informent les leurs. On passe ainsi à un mode de transmission selon lequel chacun de ces nœuds ne peut transmettre que pendant son slot. Ceci permet d'éviter les problèmes de collisions dues au problème du terminal caché qui influent énormément sur la performance du réseau lorsque la charge augmente.

Z-MAC se veut tolérant aux erreurs de synchronisation. Au début du déploiement, Z-MAC effectue une synchronisation globale de tous les nœuds, nécessaire pour le bon fonctionnement de l'ordonnancement. Ensuite, chaque nœud exécute en local un protocole de synchronisation à "faible-coût". Ce protocole s'inspire d'une technique utilisée dans RTP/RTCP [Group 1996]. Chaque nœud transmet un message de synchronisation pour 100 messages de données transmis.

Les performance de Z-MAC sont supérieures à celles de B-MAC pendant les moyennes et les hautes charges du réseau mais sont inférieures lorsque la charge est faible [Rhee 2008]. Cependant, les auteurs de [Ahn 2006] ont montré que le débit de Z-MAC se dégrade avec le temps et tend vers celui de B-MAC. Cela est dû aux différents changements du réseau, comme par exemple l'état du canal de transmission, qui invalident l'ordonnancement résultant de l'exécution de DRAND la première fois.

2.1.3.2 Crankshaft

Crankshaft [Halkes 2007] est un protocole MAC hybride. Dans ce protocole, le temps est divisé en trames et chaque trame est composée de plusieurs slots. Il y a deux types de slots : slots de diffusion et slots unicast. Pour recevoir des données, chaque nœud choisit un slot unicast ainsi que tous les slots de diffusion durant lesquels il écoute le canal. Ainsi, Crankshaft ordonnance la réception des données plutôt que leur envoi. Un slot unicast est lui même divisé en deux : partie de résolution de la contention et partie de réception des données. Le numéro du slot choisi par chaque nœud est égal au reste de la division Euclidienne du nombre total de slots unicast,

n, par l'adresse MAC du nœud lui-même. Ainsi, un nœud voulant transmettre un message au nœud suivant, connait exactement le numéro de slot dans lequel il doit transmettre. Cependant, cette allocation de slot peut engendrer le choix du même slot par deux voisins. Ainsi, pour permettre aux voisins de communiquer dans ces conditions, les nœuds sont autorisés à transmettre pendant leurs slots. Ils repassent en mode réception s'ils perdent l'accès au canal.

Crankshaft exige que tous les nœuds soient synchronisés. Cependant les auteurs ne proposent pas un algorithme de synchronisation, ils suggèrent l'utilisation du coordinateur du réseau pour tout synchroniser ou utiliser un algorithme distribué comme GSA [Li 2005]. Une conséquence de cette synchronisation est qu'un récepteur n'a pas besoin de se réveiller pendant toute la durée de son slot, mais uniquement lorsque la phase de contention se termine. Un nœud voulant transmettre un message vers un autre, choisit un instant dans la partie de résolution de contention du slot récepteur. Juste avant cet instant là, il se réveille et teste si le canal est libre. Si tel est le cas, il transmet un préambule assez long pour que le récepteur la détecte en se réveillant. Après, il envoie son message. Si le canal n'est pas libre, il doit attendre le prochain slot pour réessayer de transmettre. La Fig. 2.6 donne un exemple d'opération de Crankshaft dans un slot. Dans cet exemple, "Sender A" et "Sender B" ont un message à transmettre vers "Receiver". "Sender A" choisit un instant inférieur à celui de "Sender B". Il se réveille et teste le canal. Puisque aucun nœud n'est en train de transmettre, il transmet un préambule. Lorsque "Sender B" se réveille, le canal sera occupé et donc il annule sa transmission. Le "Receiver" détecte le canal occupé au moment de son réveil, à cause du préambule, et reste donc actif pour recevoir le message.

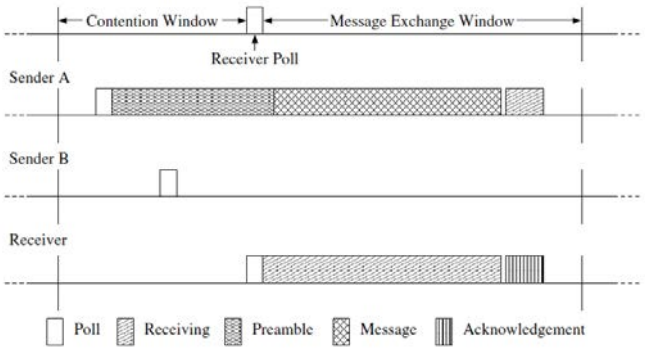


FIGURE 2.6 – Exemple d'opération de Crankshaft dans un slot [Halkes 2007].

Cranshaft obtient de bons résultats en terme de consommation d'énergie mais au prix d'un débit réduit (donc performances médiocres en terme de délai de bout en bout et de taux de succès de transmission). Le protocole est trop rigide et ne permets pas une adaptation en fonction du trafic.

2.1.3.3 Funneling-MAC

Funneling-MAC [Ahn 2006] est un protocole MAC hybride qui traite le problème de congestion et de perte de paquets observé par les nœuds proches du puits. Le protocole utilise TDMA dans cette région là et CSMA ailleurs. TDMA est géré par le puits. Tous les nœuds utilisent CSMA par défaut, à moins de recevoir un beacon du puits les informant qu'ils doivent utiliser TDMA. Ce beacon sert aussi à les synchroniser. La puissance de transmission du beacon est variable en fonction de l'intensité du trafic reçu. En effet, le puits mesure continuellement l'intensité du trafic reçu. L'ordonnancement des slots à allouer est effectué aussi par le puits en observant les différentes routes prises par les paquets et en allouant les slots aux nœuds d'une route en fonction du trafic y circulant.

2.1.4 Protocoles avec gestion de la QdS

2.1.4.1 I-EDF

I-EDF [Caccamo 2002] est un protocole MAC qui offre des garanties temps réel dures en terme de délai de bout-en-bout. Le protocole adopte une structure cellulaire du réseau où chaque cellule est assimilée à un hexagone et utilise une fréquence différente de celles de ses voisins. Dans I-EDF, le temps est divisé en trames et tous les nœuds du réseau sont synchronisés. L'accès au canal pour les nœuds appartenant à une même cellule se fait selon l'ordonnancement EDF [Liu 1973] ; le paquet ayant la plus courte échéance est transmis le premier. Cela suppose que tous ces nœuds connaissent à tout moment les paquets et leurs échéances qui seront transmis à l'intérieur d'une cellule, ce qui est possible puisque les auteurs supposent que le trafic échangé est ou bien périodique ou bien apériodique mais une période de génération minimale connue. Un routeur est placé au milieu de chaque cellule pour assurer la communication inter-cellulaire. Il est capable d'émettre et de recevoir sur des fréquences différentes au même temps (utilisation de deux transceivers). Des trames prédéfinies sont allouées à ces routeurs pour assurer la communication. Dans chaque trame le routeur transmet sur la fréquence du prochain saut. Notons que tous les routeurs transmettent sur une des six directions possibles, définies par l'hexagone, et changent de direction d'une façon cyclique.

I-EDF suppose un déploiement hexagonal et précis du réseau ainsi qu'une synchronisation globale de tout le réseau afin d'offrir des garanties temps réel. Ces restrictions supplémentaires limitent son champ d'application. I-EDF n'a pas de considération pour la consommation d'énergie.

2.1.4.2 Q-MAC

Le protocole Q-MAC [Liu 2005] utilise l'ordonnancement local (intra-nœud) pour sélectionner le prochain paquet à transmettre et un ordonnancement inter-nœud pour arbitrer l'accès au canal de transmission. Le protocole adopte un système multi-files

d'attentes (cinq files) pour classifier les paquets selon leur priorité et se base sur les algorithmes MAX-MIN [Bertsekas 1992] équitable et PGPS [Parekh 1993] pour la sélection du prochain paquet à servir. La classification se fait selon la priorité du paquet du point de vue application, le nombre de sauts traversés, l'énergie restante du nœud et la charge pondérée de la queue. L'ordonnancement inter-nœud utilise un protocole d'accès aléatoire avec une gestion souple de la priorité (en anglais "Loosely Prioritized Random Access"). Le temps est divisé en parties actives et parties inactives avec un rapport cyclique fixe, comme dans S-MAC. La partie active est divisée en trames. Une trame représente un échange RTS-CTS-DATA-ACK et consiste en une période de contention (PC) et une période de transmission (PT). Durant la PC les nœud ayant des données à transmettre essayent d'établir un échange RTS-CTS afin de réserver le canal, *i.e.* ils transmettent un message RTS et attendent la réception du CTS, le premier qui établit cet échange gagne l'accès au support. La PC est divisée en cinq sous périodes de contention, chacune attribuée à une priorité ; la première étant attribuée à la plus haute priorité. De cette façon les nœuds qui ont des messages plus urgents échangent des trames RTS-CTS les premiers.

Le fonctionnement de Q-MAC suppose une synchronisation globale du réseau. De plus, l'utilisation d'un rapport cyclique fixe le rend peu adapté aux variations du volume et de la nature du trafic.

2.1.4.3 RL-MAC

RL-MAC [Liu 2006] est un protocole MAC inspiré de S-MAC et qui utilise la technique d'apprentissage par renforcement [Sutton 2005] afin de déterminer la meilleure taille de la période active. Pour un nœud donné, l'adaptation de la taille de la période active se base sur les observations des paquets correctement reçus et transmis par lui ainsi que le nombre d'échecs de transmission de ses voisins. Ainsi, l'objectif est de minimiser la consommation d'énergie et le nombre de paquets perdus à cause du problème de sommeil trop précoce ("early sleeping"). RL-MAC implémente trois classes de trafic. La différenciation entre eux se fait en variant la taille de la CW pour chaque type de trafic.

L'avantage de RL-MAC est l'adaptation de la période d'activité en fonction du trafic. Cependant, la complexité de l'algorithme d'adaptation peut constituer un handicap pour son implémentation sur les capteurs sans fil.

2.1.5 Standards

2.1.5.1 IEEE 802.15.4

Description générale : IEEE 802.15.4 est une suite de protocoles de communication à bas débit qui fait partie des réseaux locaux personnels. Il a été conçu pour garantir une facilité d'installation, un maximum de fiabilité de transfert de données, une opérabilité à court terme, un très faible coût et une durée de batterie

raisonnable. Le design du protocole reste aussi simple et très flexible.

Le réseau supporte deux types de périphériques, les FFD (Full-Function Devices) et les RFD (Reduced-Function Devices). Les premiers implémentent toutes les primitives (fonctions) définies par la norme. Ils communiquent entre eux et avec les RFD alors que ces derniers ne communiquent qu'avec des FFD. Un RFD est conçu pour les applications simples comme les interrupteurs d'électricité ou les capteurs infrarouges et n'ont généralement pas besoin de transmettre beaucoup de données. En conséquence, les RFD sont implémentés avec le minimum de ressources possibles.

Un FFD peut être soit un coordinateur, un coordinateur du PAN ou juste un simple périphérique. Le coordinateur du PAN est un FFD qui démarre le réseau et fournit tous ses paramètres fonctionnels. Il est aussi responsable de la gestion du réseau (synchronisation, requêtes d'association, *etc.*). Un coordinateur est un FFD qui joue le rôle d'un routeur afin d'étendre le réseau. Il peut fournir les mêmes services de gestion que le coordinateur du PAN.

La sous couche MAC : La sous couche MAC supporte deux modes d'accès qui peuvent être sélectionnés par le coordinateur du PAN :

- Le mode sans Beacon, dans lequel l'accès est contrôlé par le protocole CSMA/CA non slotté.
- Le mode avec Beacon, dans lequel des beacons [4] sont régulièrement émis par le coordinateur du PAN ainsi que d'autres coordinateurs pour synchroniser les nœuds associés et pour identifier le PAN. Dans ce cas, une structure de supertrame est définie entre deux transmissions successives de beacons. L'accès dans ce mode est commandé essentiellement par le protocole CSMA/CA slotté mais il implémente également un mécanisme d'allocation de tranches [5] de temps dites (GTS) pour les nœuds qui nécessitent des services garantis.

La structure de la supertrame est définie par Fig. 2.7

La supertrame est divisée en deux parties, une partie active où les nœuds peuvent émettre des données, cette partie est toujours divisée en 16 tranches de temps (slots), et une partie inactive où ils restent en état de veille. Elles sont définies par deux paramètres qui sont *l'intervalle Beacon (BI)* et *la durée de la supertrame (SD).*

$$BI = aBaseSuperframeDuration \cdot 2^{BO} \qquad \text{for} \quad 0 \leq BO \leq 14 \tag{2.1}$$

$$SD = aBaseSuperframeDuration \cdot 2^{SO} \qquad \text{for} \quad 0 \leq SO \leq BO \tag{2.2}$$

Dans les deux équations 2.1 et 2.2, *aBaseSuperframeDuration* est une constante, c'est la durée minimale de la supertrame correspondant à $BO = 0$. Elle est fixée à 960 symboles ou 15.36 ms (considérant un débit de 250 Ko/s) selon le standard [6]. BO est l'ordre du Beacon et SO est l'ordre de la supertrame. Il est à noter que puisque le nombre de tranches est constant dans une supertrame (égale à seize) et

4. ce sont des trames spéciales envoyées périodiquement
5. tranche est la traduction du terme anglais "slot"
6. un symbole correspond à 4 bits

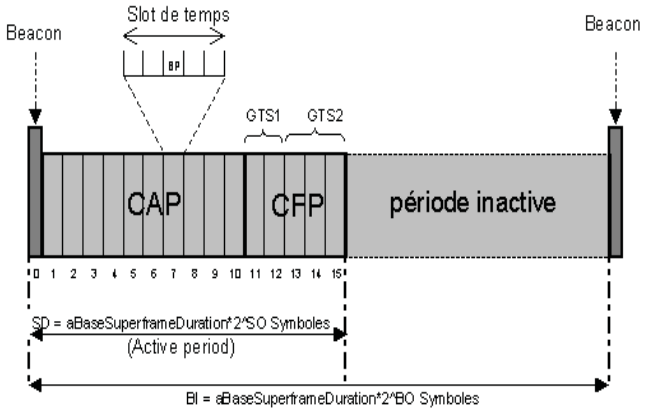


FIGURE 2.7 – Structure de la supertrame

que la taille de celle ci est variable, la taille d'une tranche de temps est aussi variable. Elle est égale à 60 symboles si $BO = SO = 0$

La partie active de la supertrame peut elle-même être divisée en deux périodes : période où le support de transmission est partagé, les nœuds utilisent alors le protocole CSMA/CA slotté pour disputer l'accès au canal de transmission. Elle est appelée CAP (période pour l'accès partagé). La deuxième période est optionnelle et est activée par un coordinateur suite à la demande d'un nœud pour lui allouer des GTS (tranches de temps réservées pour un nœud), c'est la partie CFP (période d'accès libre).

Le standard supporte les topologies en étoile, en arbre et maillée. Les coordinateurs peuvent aussi à leur tour fournir des services de synchronisation pour d'autres coordinateurs et RFD. Ainsi, un mécanisme d'ordonnancement de beacons doit être mis en place pour éviter la collision des beacons entre coordinateurs. Le standard ne définie pas un tel algorithme mais il impose comme contrainte que BO et SO soient égales pour toutes les supertrames du PAN.

CSMA/CA : Le standard IEEE 802.15.4 [IEEE-TG15.4 2006] utilise le protocole CSMA/CA pour l'accès partagé au support. Les deux variantes du protocole sont illustrées par la figure 2.8.

Dans les deux variantes, le protocole utilise une unité basique de temps appelée BP (Backoff Period) qui correspond à 20 symboles[7]. Pour la supertrame par exemple, chaque tranche de temps (" time slot ") est constituée par plusieurs BP. Le nombre de BP dans une tranche de temps est donné par l'équation (3).

$$N = 3 \cdot 2^{SO} \tag{2.3}$$

7. 0.32 ms pour un débit de 250 Ko/s

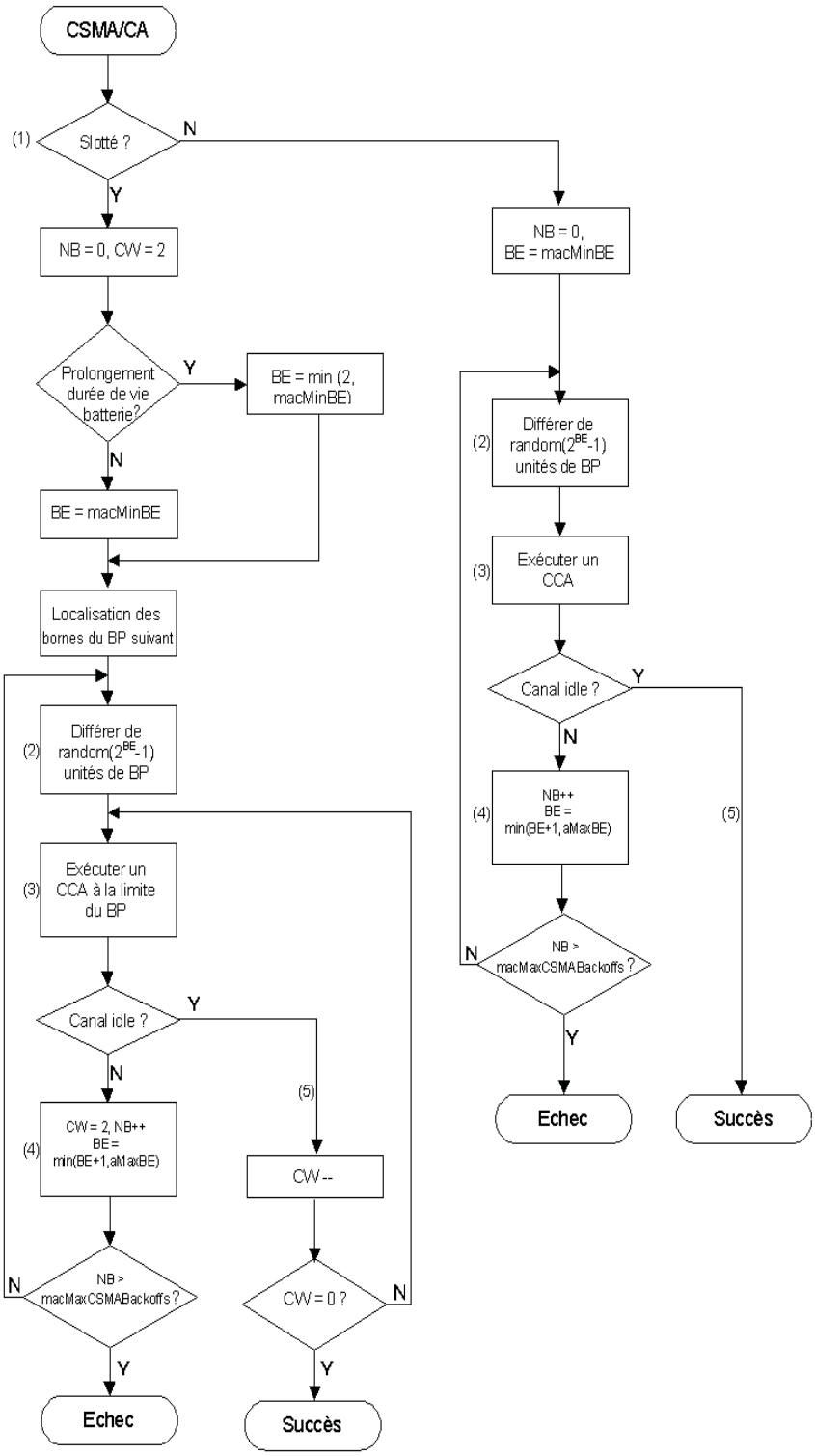


FIGURE 2.8 – Le protocole CSMA/CA

Dans le protocole CSMA/CA slotté[8], chaque opération (accès au canal, CCA, décompte de backoff) ne peut se faire qu'au début d'un BP. De plus, les intervalles de BP doivent être alignés avec ceux de la supertrame. Dans celui non slotté, les BP d'un nœud sont totalement indépendants de ceux des différents autres nœuds.

CSMA/CA dépend essentiellement de trois variables :

- BE (exposant du backoff) permet le calcul du délai de BP. Ce délai correspond au temps aléatoire (compris entre 0 et $(2^{BE} - 1)$) que doit attendre le nœud avant d'effectuer la procédure de CCA[9]. La valeur minimale de BE est appelée $macMinBE$. Elle est égale à trois par défaut.
- CW (fenêtre de compétition). Représente, le nombre de fois où le canal est détecté libre (idle) avant d'accéder au canal (fixé à deux par le standard).
- NB (nombre de backoffs) représente le nombre de fois où le nœud tente de transmettre avant de déclarer un échec de transmission.

CSMA/CA slotté se déroule en 5 étapes :

1. Initialisation des valeurs de NB à 0, CW à 2 et BE à macMinBE ou à 2 en fonction du paramètre du prolongement de la durée de vie de la batterie et repérage des bornes de la prochaine BP.
2. Appel de la procédure de CCA à l'expiration du délai.
3. Exécution de la procédure CCA si le canal est occupé il passe à l'étape (4) sinon il passe à l'étape (5).
4. Le canal est occupé : la valeur de CW est réinitialisée à 2 et celle de BE est incrémentée (la valeur maximale est $aMaxBE$ qui est égale à cinq par défaut). Si le canal est détecté occupé pendant $macMaxCSMABackoffs$ (valeur par défaut égale à quatre) la transmission échoue et l'algorithme signale l'échec à sa couche MAC.
5. Canal libre, la valeur de CW est décrémentée. Si elle atteint 0 alors le nœud tente une transmission. Sinon, la procédure de CCA est exécutée pour la deuxième fois. Cette valeur est mise à deux pour permettre la transmission des messages d'acquittement.

Les étapes de CSMA/CA non slotté sont presque ceux du slotté à l'exception que celui-ci n'utilise pas la variable CW et ne fait pas de repérage des bornes de la BP suivante puisqu'il n'utilise pas la structure de la supertrame.

2.1.5.2 WirelessHart

WirelessHART [HartComm 2007] est un standard ouvert de réseaux sans fil développé par la "HART Communication Foundation". Il vise plutôt des applications industrielles.

8. Le terme slotté est en rapport avec la BP, à ne pas confondre avec le slot correspondant aux 16 tranches de temps
9. CCA : Clear channel Assessment, Détection de l'activité du canal

WirelessHART définit toute la pile protocolaire de communication, *i.e.* de la couche physique à la couche application. La couche physique est celle de la bande 2.4 GHz définie par le standard IEEE 802.15.4.

WirelessHART utilise un réseau maillé non étoilé dans lequel chaque station impliquée sert simultanément de source de signal et de répéteur (Fig. 2.9). La source envoie un message à son voisin le plus proche qui le transmet à son tour à un de ses voisins jusqu'à ce qu'il atteigne la station du récepteur final. De plus, des chemins alternatifs sont définis dans la phase d'initialisation. Si le message ne peut pas être transmis sur un chemin particulier, par exemple en raison d'un obstacle ou d'un répéteur défectueux, il est automatiquement dévié sur un chemin alternatif. Ainsi, outre le fait d'étendre la portée du réseau, le réseau en maille non étoilé fournit des chemins de communication redondants afin d'améliorer la fiabilité.

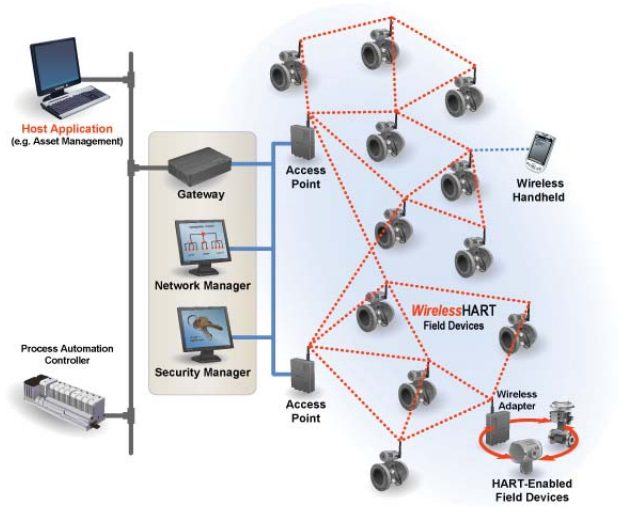


FIGURE 2.9 – Exemple d'un réseau WirelessHART [HartComm 2007].

La communication du réseau utilise la technique TDMA avec une taille de slot égale à 10 ms. Cependant, quelques slots sont réservés pour un accès partagé utilisant le protocole CSMA/CA.

Afin d'éviter les interférences, WirelessHART utilise également la modulation FHSS ("Frequency Hopping Spread Spectrum Modulation"). Les 15 canaux définis dans la norme IEEE802.15.4 sont utilisés en parallèle ; WirelessHART utilise la modulation FHSS pour effectuer le saut entre les canaux. Ainsi, les canaux déjà utilisés par d'autres réseaux ou qui souffrent de problèmes d'interférence importants sont mis sur une liste "noire" afin d'éviter des collisions avec d'autres systèmes de communication sans fil.

Toutes les opérations de création du réseau, de synchronisation, d'ordonnancement des slot, de maintenance, *etc* sont gérés par le gestionnaire du réseau.

La combinaison d'un slot toutes les 10 ms et des 15 canaux permet d'avoir 1500 slots par seconde à ordonnancer.

WirelessHART est un protocole rigide. Au delà des réseaux industriels bien définis, son application dans d'autres environnements est difficile.

2.1.6 Discussion

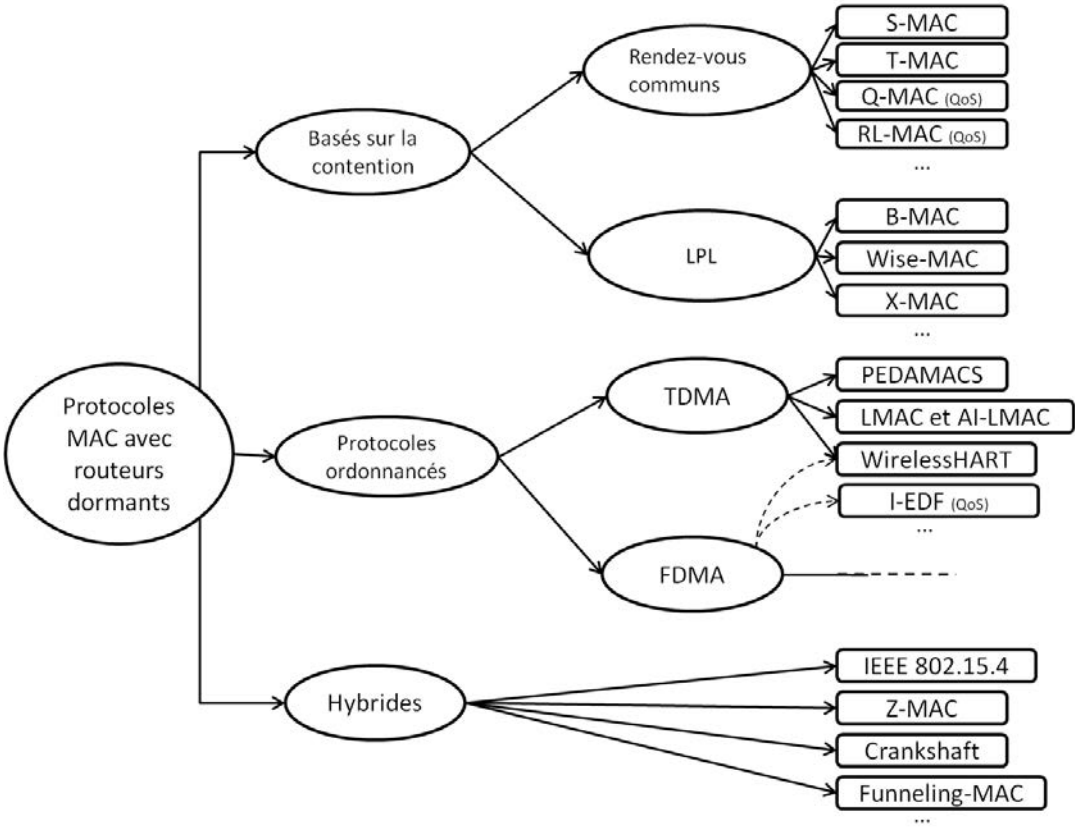


FIGURE 2.10 – Classification des protocoles MAC étudiés. Le terme (QoS) indique que le protocole supporte la QdS. WirelessHART et I-EDF incluent un ordonnancement de type FDMA

Une quantité de travail importante à été produite sur la couche MAC dans les réseaux de capteurs sans fil ces dernières dix années. Les dates de publication indiquent que les premiers travaux se sont basés sur les protocoles synchronisés comme TDMA ou les rendez-vous communs. Ensuite, des protocoles asynchrones tels que B-MAC ont vu le jour. Enfin, constatant les forces et les faiblesses de chaque type de protocole, des versions hybrides ont été développés [Bachir 2010]. Aujourd'hui la tendance est aux protocoles utilisant plusieurs canaux fréquentiels [Slimane 2010].

La figure 2.10 présente une classification des protocoles étudiés selon leur nature. On distingue principalement trois familles. Premièrement, on a les protocoles qui basés sur la contention. Ces protocoles peuvent être classés en deux catégories. Dans la première, dite à Rendez-vous commun, synchronize tout ou une partie de

l'ensemble des nœuds du réseau afin qu'ils se réveillent et se rendorment aux même instants. La deuxième catégorie se base sur la technique d'écoute à faible puissance LPL. Deuxièmement, on a les protocoles ordonnancés qui utilisent soit la technique TDMA ou celle FDMA. Finalement, on a les protocoles hybrides qui utilisent des techniques issues des deux autres familles.

Du point de vue objectif recherché, la majorité des protocoles développés se sont focalisés sur l'optimisation de la consommation d'énergie indépendamment de la charge du réseau. Cependant, dans les dernières années, les chercheurs se sont davantage intéressés à la QdS et au compromis énergie-débit.

Pour la gestion de la QdS, l'approche purement ordonnancée, à la manière de I-EDF ou PEDAMACS, est de moins en moins utilisée. Ainsi, deux tendances se dégagent au niveau de la conception des protocoles MAC [Yigitel 2011]. La première se base sur la contention avec l'adoption de l'approche de rendez-vous commun. CSMA/CA est le protocole d'accès le plus utilisé dans ce cas. La seconde est l'approche hybride, à la manière de Z-MAC. Dans les deux cas, la différenciation de service se base principalement sur la variation des paramètres du protocole d'accès en fonction de la priorité du trafic (fenêtre de contention, espace inter-trames (IFS), probabilité de sélection d'un slot, valeur de l'exposant du backoff, *etc*).

2.2 Protocoles de routage

Les protocoles de routages développés pour les réseaux de capteurs sans fil peuvent être classés en trois catégories ; basés sur une architecture plate, hiérarchique et basés sur la localisation. Nous détaillons dans les trois premières sous-sections ces catégories à travers des exemples de protocoles de routage. Ensuite, nous analysons dans la sous-section d'après quelques protocoles de routage avec gestion de la QdS. Après, nous décrivons les protocoles de routage du standard ZigBee ainsi que celui proposé par le groupe de travail ROLL de l'IETF [IETF 2011]. Enfin, nous proposons dans la dernière sous-section une analyse de l'état de l'art des protocoles de routage.

2.2.1 Protocoles basés sur une architecture plate

Les protocoles basés sur une architecture plate sont souvent des protocoles "orientés-données". A la différence des protocoles basés sur l'adressage (chaque nœud ayant un identifiant unique), ces protocoles s'intéressent à des requêtes de type "zone où la température est supérieure à une valeur donnée" au lieu de "température du nœud X" (avec X l'adresse du capteur). Ces protocoles partent du principe que les capteurs forment un réseau très dense et qu'il est difficile d'attribuer une adresse unique à chacun d'entre eux. Nous détaillerons quelques exemples de ces protocoles.

2.2.1.1 Gossipping

Ce type de protocole est une dérivation du protocole d'inondation qui résout le problème d'implosion de messages. Ce problème résulte du fait que plusieurs nœuds peuvent rediffuser le même message. Ainsi, un nœud, peut obtenir la même copie du même message N fois, avec N le nombre de ses voisins.

Comme son nom l'indique (bavardage en français), il est inspiré par le principe de diffusion de rumeurs dans les réseaux sociaux. A la réception d'un message, chaque nœud choisit un de ses voisins, au hasard, pour relayer le message et ainsi de suite [Hedetniemi 1988]. Au bout d'un moment, tous les nœuds recevront une copie du message. Le protocole de bavardage permet de borner la pire charge du réseau. Cependant, il augmente la latence de propagation des messages à tous les nœuds du réseau.

2.2.1.2 La famille SPIN

Kulik et al. ont proposé une famille de protocoles appelée SPIN (Sensor Protocols for Information via Negotiation), reposant sur un modèle de négociation afin de propager l'information dans un réseau de capteurs [Kulik 2002]. Le but de SPIN est de pallier les problèmes de l'inondation, qui sont :

- L'implosion due à la duplication inutile des réceptions d'un même message.
- Le chevauchement lié au déploiement dense des capteurs. En utilisant l'inondation, les capteurs d'une zone émettront tous la même donnée (ou presque).
- L'ignorance des ressources, car l'inondation ne prend pas en considération les ressources des nœuds.

Ces trois problèmes affectent grandement la durée de vie et les performances du réseau. Pour les résoudre, SPIN adopte deux principes :

- La négociation : pour éviter le problème d'implosion, SPIN précède l'émission d'une donnée par sa description, en utilisant la notion de métadonnées. Le récepteur aura le choix par la suite d'accepter la donnée ou non. Ce mécanisme permet aussi de régler le problème de chevauchement.
- L'adaptation aux ressources : d'une manière continue, les nœuds contrôlent leur niveau d'énergie. Le protocole SPIN adapte son exécution suivant l'énergie restante du capteur, et modifie en conséquence le comportement du nœud.

Les communications dans SPIN se font en trois étapes :

- Lorsqu'un nœud veut émettre une donnée, il émet d'abord un message ADV contenant une description de la donnée en question. Ceci est illustré par les figures (Fig. 2.11(a) et Fig. 2.11(d)).
- Un nœud recevant un message ADV, consulte sa base d'intérêt. S'il est intéressé par cette information, il émet un message REQ vers son voisin. Ceci est illustré par les figures (Fig. 2.11(b) et Fig. 2.11(e)).
- En recevant un message REQ, l'émetteur transmet à l'intéressé la donnée sous forme d'un message DATA. Ceci est illustré par les figures (Fig. 2.11(c) et

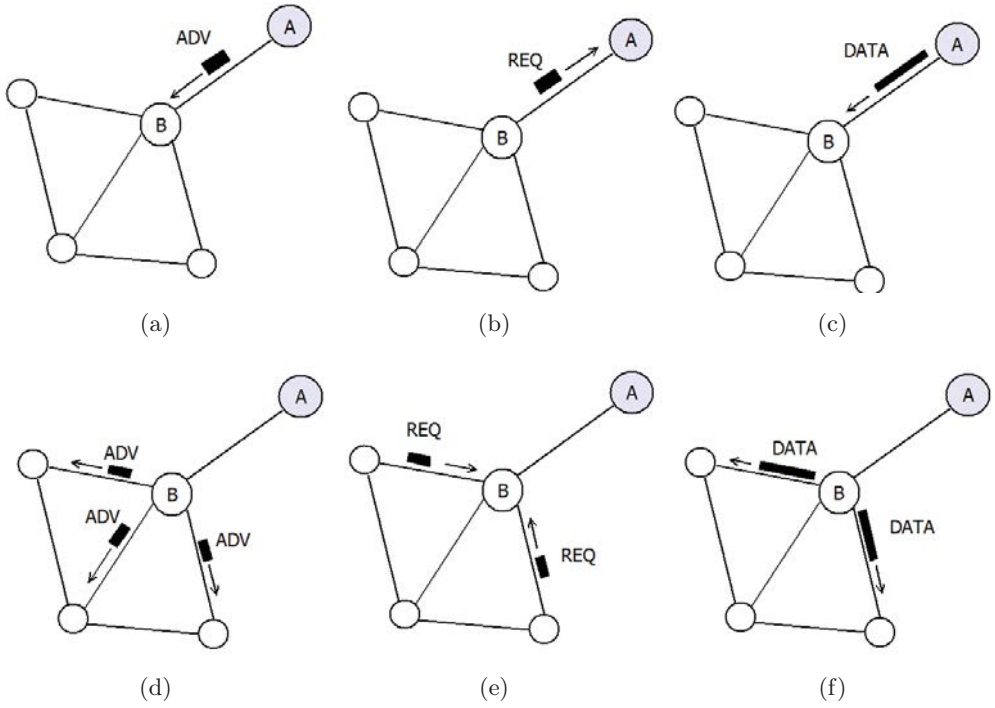


FIGURE 2.11 – Exemple d'opération du protocole SPIN.

Fig. 2.11(f)).

Ainsi, la propagation des données jusqu'au puits est faite selon ce principe. Notons qu'à chaque étape, plusieurs nœuds peuvent envoyer des REQ au nœud émettant les données. Celui-ci envoie une copie des données à chacun d'entre eux. Ce protocole est appelé SPIN-PP (pour SPIN point-à-point).

Plusieurs variantes de ce protocole ont été proposées pour pallier les défauts de SPIN-PP. On retrouve ainsi SPIN-EC (pour SPIN "Energy Consumption awarness), SPIN-BC (pour SPIN for Broadcast networks) et SPIN-RL (pour SPIN with reliability).

SPIN-PP ne résout pas totalement le problème d'ignorance de ressources. SPIN-EC résout ce problème par rapport à la consommation d'énergie. A chaque fois qu'un ADV est reçu, le nœud intéressé par la donnée calcule le budget énergétique de l'opération (Envoi de REQ, réception et retransmissions). Si son énergie actuelle permet d'effectuer cette opération, il envoie une REQ. Sinon il se tait.

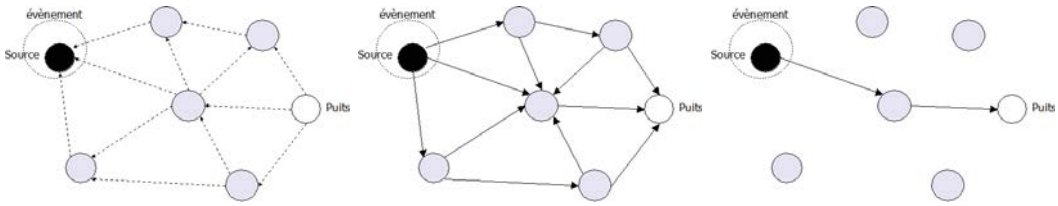
Un autre défaut de SPIN-PP est que lors de la réception de plusieurs REQ pour la même donnée, le nœud envoie une copie pour chacun des intéressés. Or le canal sans fil est diffuseur de nature. SPIN-BC exploite cette propriété. Si un nœud est intéressé par une donnée, il choisit une période aléatoire avant de transmettre le REQ. Pendant cette période il continue d'écouter le canal. S'il reçoit un REQ d'un autre nœud pour la même donnée, il annule sa transmission et attend la réception du message.

SPIN-RL propose un mécanisme de fiabilité pour SPIN-BC. Si un nœud a reçu un ADV mais pas la donnée, il demande à ses voisins de lui envoyer une copie de la donnée s'ils l'ont reçue.

SPIN réduit considérablement la consommation d'énergie par rapport à l'inondation. L'amélioration est de l'ordre de 70% pour SPIN-PP. Cependant, il a une latence plus élevée pour diffuser les données dans le réseau à cause du mécanisme d'échange avant la transmission de la donnée.

2.2.1.3 Diffusion dirigée

La diffusion dirigée (DD) [Intanagonwiwat 2000] se base sur le modèle publish/subscribe. Le puits diffuse une requête (appelée intérêt) afin d'interroger le réseau sur une donnée particulière. Il opère en quatre phases pour construire les routes entre le puits et les capteurs concernés par l'intérêt diffusé. Les quatres phases sont : (1) propagation des intérêts, (2) établissement des gradients, (3) propagation des données et (4) renforcement des chemins.



(a) Propagation des intérêts (b) Envoie des données selon les gradients établis (c) Renforcement d'un chemin

FIGURE 2.12 – Exemple d'opération du protocole DD.

Lorsqu'un puits requiert une donnée du réseau, il propage un intérêt, contenant sa description ainsi que le débit d'information désiré. L'intérêt est diffusé dans tout le réseau en utilisant l'inondation. Initialement, le puits spécifie un grand intervalle (donc un débit faible), dans un but d'exploration. Cela permet d'établir les gradients et de découvrir d'éventuelles sources, sans pour autant encombrer le réseau. La figure 2.12(a) illustre la propagation de l'intérêt.

A la réception, d'un intérêt, chaque nœud met à jour son cache d'intérêts. Le cache d'intérêts contient plusieurs champs y compris l'estampille, le gradient, l'intervalle et la durée. L'estampille indique l'instant local de réception de l'intérêt. Le gradient indique le nœud qui a envoyé l'intérêt. Ce champ est utilisé pour former les chemins inverses vers le puits. Chaque entrée dans le cache a une durée de vie indiquée par le champ durée. L'intervalle spécifie le débit des données. Après la réception de l'intérêt, le nœud diffuse le message à ses voisins.

Chaque nœud vérifie si ses observations correspondent à l'intérêt diffusé. Si c'est le cas, il devient un nœud source. Il commence ainsi à envoyer les données à travers ses gradients, comme le montre la Fig. 2.12(b). Notons qu'un nœud source peut avoir

plusieurs gradients puisqu'il peut recevoir l'intérêt de plusieurs voisins. Lorsqu'un nœud reçoit une donnée, il recherche un intérêt équivalent dans son cache. Si aucune entrée n'est trouvée, le paquet est supprimé. Dans le cas contraire, en consultant la liste des gradients, le nœud relaie la donnée vers ses voisins.

Puisqu'un nœud source peut avoir plusieurs gradients, les données peuvent emprunter plusieurs chemins. Ainsi, lorsque le puits reçoit les premières données, il peut décider de renforcer un de ces chemins et d'augmenter le débit de captage. Le choix du chemin est fait selon plusieurs règles comme par exemple le nombre de messages reçus, la qualité du lien, le délai le plus court, *etc.* Le message de renforcement est envoyé au nœud choisi, qui à son tour le relaie à travers le chemin choisi vers la source. La figure 2.12(c) illustre le chemin renforcé par le puits.

Dans le cas de panne d'un lien (perte de paquets, débit réduit, etc.) le puits peut envoyer un renforcement négatif sur le chemin en panne en spécifiant le débit de base (exploratoire), et en procédant à un renforcement positif d'un chemin alternatif.

Le principal défaut de DD est l'opération d'inondation des intérêts qui conduit à une consommation importante d'énergie.

2.2.2 Protocoles hiérarchiques

2.2.2.1 LEACH

LEACH [Heinzelman 2000] est un protocole de routage qui vise l'optimisation de la consommation d'énergie dans le réseau en formant des clusters. Ainsi, l'objectif de LEACH est de sélectionner dynamiquement des nœuds pour agir comme des têtes de clusters et former des clusters dans le réseau. Les autres nœuds rejoignent alors un des clusters formés. Ces nœuds transmettent leurs données à la tête du cluster, qui se chargera de les agréger et les envoyer directement au puits. Une des hypothèses dans LEACH est que tous les nœuds peuvent joindre le puits en un seul saut. Les têtes de clusters sont échangées périodiquement pour que l'énergie consommée en communiquant avec le puits soit distribuée entre tous les nœuds du réseau.

LEACH opère en deux phases d'une manière cyclique : (1) phase de création et (2) phase d'équilibre. Pendant la phase de création les têtes de clusters sont sélectionnées, les clusters sont formés et l'ordonnancement de la communication est déterminé. Notons que les membres du même cluster utilisent un protocole de type TDMA pour communiquer. Les têtes de clusters utilisent CDMA (Code Division Multiple Access) pour communiquer avec le puits. Pendant la phase d'équilibre, les membres de clusters commencent à capter et à transmettre les données aux têtes de clusters, qui vont agréger ces données et transmettre le résultat au puits.

Au début de chaque cycle, les têtes de cluster sont choisies aléatoirement. Premièrement, un nœud choisit aléatoirement un nombre entre 0 et 1. Si ce nombre est inférieur à un certain seuil T(n), le nœud est élu comme tête de cluster. T(n) est

donné par l'équation 2.4.

$$T(n) = \begin{cases} \frac{P}{1-P(r \bmod (1/P))}, & \text{if } n \in G \\ 0, & \text{Sinon.} \end{cases} \quad (2.4)$$

Dans (2.4), P est le pourcentage désiré de nœuds qui deviendront des têtes de clusters, r est le cycle actuel et G est l'ensemble de nœuds qui n'ont pas été élu comme têtes de clusters dans les derniers $1/P$ cycles. Une fois l'élection faite, les nouvelles têtes de clusters annoncent leurs élections aux voisins, qui vont maintenant choisir un cluster. A la fin d'association des nœuds aux clusters, les têtes de clusters ordonnancent les communications des fils. Notons que pendant la phase de création les nœuds utilisent le protocole CSMA/CA pour communiquer.

LEACH améliore la consommation d'énergie d'un facteur de 4 à 8 par rapport aux architectures plates. Cependant, certaines des hypothèses sur lesquelles il se base limitent son application. En effet, LEACH suppose que tous les nœuds sont capables d'atteindre le puits en un seul saut. En plus, un nœud doit implémenter trois couches MAC différentes (CSMA, TDMA et CDMA).

2.2.2.2 PEGASIS

PEGASIS [Lindsey 2002] propose une amélioration de LEACH en éliminant le coût additionnel généré par l'opération de création des clusters. PEGASIS construit des chaînes de nœuds au lieu de clusters. Pour construire les chaînes de nœuds, chacun d'entre eux, sélectionne le voisin le plus proche comme prochain saut dans la chaîne. PEGASIS suppose que les nœuds ont une connaissance globale du réseau. La construction de la chaîne commence alors des nœuds les plus loin par rapport au puits. Ainsi, au lieu de former des clusters, chaque nœud maintient uniquement la liaison entre son prédécesseur et son successeur dans la chaîne.

La communication dans la chaîne est réalisée d'une manière séquentielle de telle sorte que chaque nœud agrège les données de son voisin avec les siennes et ce jusqu'à ce que toutes les données soient agrégées dans la tête de la chaîne. Celui-ci contrôle l'ordre de la communication des nœuds en utilisant un jeton.

PEGASIS améliore la performance énergétique de LEACH de 100 à 300% au prix de délais additionnels. Son application reste aussi limitée que LEACH voire même plus, puisqu'il suppose une connaissance totale du réseau.

2.2.3 Protocoles basés sur la localisation

L'information sur la localisation est essentielle dans plusieurs applications de WSN. En effet, puisque l'intérêt d'un tel réseau est de reporter des événements physiques, la connaissance de la position des capteurs est nécessaire pour associer les observations avec leur position géographique.

Afin de déterminer leurs positions, les nœuds utilisent ou bien des GPS ou bien des techniques de localisation. Ainsi, puisque l'information sur la localisation est

nécessaire et disponible, elle peut être exploitée pour le routage. On parle alors de protocoles de routage géographique.

Pour effectuer la décision de routage, un nœud a besoin de sa position géographique, de celle de la destination et de celles de ses voisins. La position de la destination est souvent fixe dans les WSN, qui est celle du puits. La position des voisins est déterminée à travers un échange de messages HELLO (bonjour).

Nous présentons ici les deux principales techniques de routage géographique à savoir (1) les algorithmes gloutons et (2) le routage en face. Le troisième paragraphe détaille un exemple de routage géographique hybride, le GEDIR-FACE2.

2.2.3.1 Algorithmes gloutons (greedy Algorithms)

Un algorithme glouton est un algorithme qui suit le principe de faire, étape par étape, un choix optimum local, dans l'espoir d'obtenir un résultat optimum global [10].

L'algorithme glouton [Finn 1987] (du même nom que son type) est l'algorithme de routage le plus simple. Si un nœud N veut transmettre à une destination D, il compare la position de tous ses voisins à D et choisit le plus proche de D. Ce protocole ne nécessite ni une infrastructure, ni un graphe organisé d'une certaine manière. Malgré sa simplicité, l'algorithme glouton a quelques défauts. Le plus important entre eux étant qu'il peut ne pas trouver, dans certains cas, le chemin vers la destination, même s'il en existe un. Ceci arrive généralement dans les zones à faibles densités ou lorsqu'il y a des trous dans le réseau. Un nœud intermédiaire se retrouve donc comme un minimum local sur le chemin source-destination.

Le routage en boussole [Kranakis 1999] est un autre algorithme glouton qui se base cette fois sur les directions au lieu de la distance. Un nœud, S, choisit comme prochain saut le voisin, M, qui a la direction la plus proche de la destination D (*i.e.* M, qui minimise l'angle $M\widehat{S}D$). Le principal défaut du routage en boussole est qu'il peut créer des boucles infinies.

Plusieurs variantes de ces algorithmes existent. GEDIR est l'un d'entre eux. Il sera décrit dans 2.2.3.3.

2.2.3.2 Face Routing

Le principal défaut des algorithmes gloutons est l'incapacité de trouver, dans certains cas, le chemin vers la destination même s'il en existe un. Ainsi, d'autres algorithmes doivent être mis en place pour assurer la convergence dans les protocoles de routage géographique. Une des solutions, le routage en face, provient de l'idée pour trouver le chemin de la sortie dans les labyrinthes ; en ne levant jamais la main droite des murs du labyrinthe. Ainsi, en imaginant le réseau comme un graphe, on peut vérifier facilement, qu'un paquet reviendra toujours à la source en traversant les bords d'une face du graphe. Avec quelques règles en plus, on peut donc forcer le paquet à visiter

10. Source : Wikipedia

toutes les faces du graphe en séquence jusqu'à atteindre la destination. Notons que le routage en face nécessite des **graphes planaires** (*i.e.* c'est un graphe où il n'y a pas de bords qui se croisent). Ainsi, la première phase du réseau consiste à construire ce graphe.

Kranakis et al. ont proposé dans [Kranakis 1999] un algorithmes basé sur la règle de la main droite appelé FACE-1. L'algorithme est décrit dans Algorithm 1. La figure

Algorithm 1: Algorithme de FACE-1

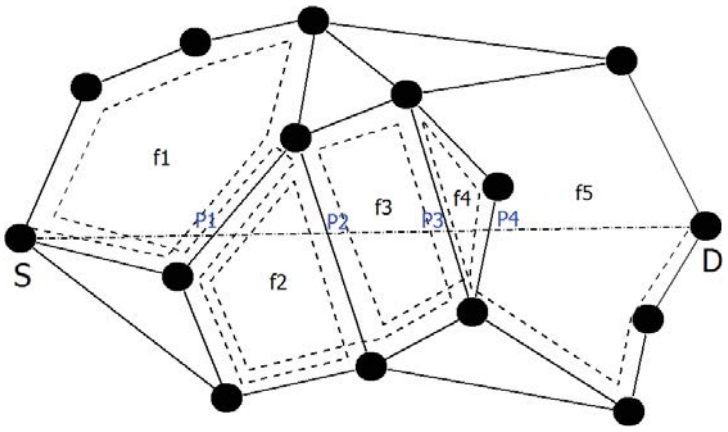
```
constant S ← source;
constant D ← destination;
constant r ← ligne du segment SD;
P ← S;
while P ≠ D do
    f ← première face qui coupe le segment PD de P à D;
    forall the bords e de f do
        if e coupe r au point P' et P' est plus proche de D que P then
            P ← P';
        end
    end
    traverser f encore jusqu'à atteindre le bord où P a été trouvé;
end
```

2.13(a) donne un exemple d'opération de FACE-1. Le trait en pointillés donne le parcours du paquet pour atteindre la destination.

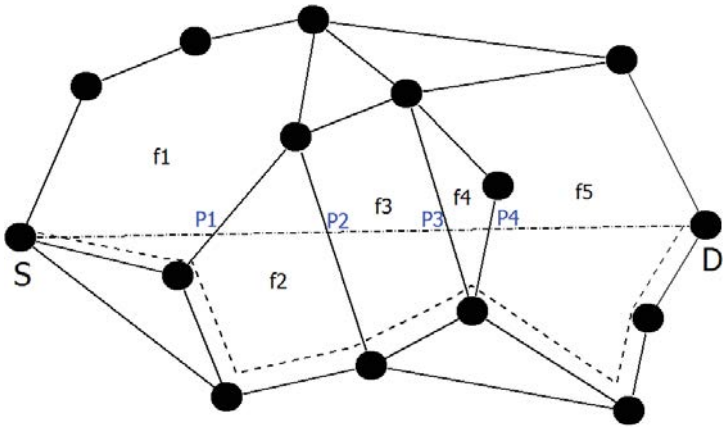
Un défaut de FACE-1 est le nombre de tours que le paquet fait à chaque face du graphe. Bose et al. ont proposé dans [Bose 2001] une amélioration pour cet algorithme qu'ils ont bapitisé FACE-2. A la différence de FACE-1, le paquet passe d'une face à une autre dès qu'il trouve un bord qui coupe le segment SD. FACE-2 améliore le nombre total de sauts dans la plupart des cas. Cependant, dans certains cas (dont un exemple présenté par les auteurs) il repasse par les mêmes nœuds plusieurs fois avant d'atteindre la destination. Il devient donc moins performant que FACE-1. L'algorithme de FACE-2 est donné par Algorithm 2. La figure 2.13(b) donne un exemple d'opération de FACE-2.

Dans tous les cas, les algorithmes de face ne sont pas très performants quand il s'agit du nombre total de sauts à faire avant d'atteindre la destination. C'est pour cela qu'il est préférable d'utiliser un algorithme glouton, puis basculer au routage de face quand le message atteint un minimum local.

Un autre problème du routage de face est que la construction des graphes planaires reste une tâche difficile à réaliser (mais pas impossible) dans la pratique, surtout dans les WSN.



(a) Exemple d'opération de FACE-1



(b) Exemple d'opération de FACE-2

FIGURE 2.13 – Illustration du protocole de routage en face. Le chemin en pointillés montre le chemin suivi par FACE 1 et FACE 2.

2.2.3.3 GEDIR and GEDIR-FACE2

GEDIR, GEographical DIstance Routing, est une amélioration de l'algorithme glouton. Lorsqu'un message atteint un minimum local, au lieu de le supprimer comme dans l'algorithme glouton, il obtient une chance supplémentaire pour échapper à ce minimum en le transmettant au voisin le plus proche de la destination autre que le nœud lui même. GEDIR améliore nettement le taux de délivrance des paquets par rapport à la version originale. Cepedant, il ne résout pas totalement le problème.

Ainsi Bose et al. ont proposé dans le même papier qui présente FACE-2, GEDIR-FACE2, une version hybride de GEDIR. L'algorithme démarre en utilisant GEDIR (puisqu'il est plus performant dans les réseaux denses) et commute à FACE-2 dès que le paquet atteint un minimum local, M. GEDIR reprend la main dès que FACE-2 trouve un nœud N, plus proche que M de la destination.

La famille de protocoles de routage géographique est certainement celle qui engendre

Algorithm 2: Algorithme de FACE-2

constant S ← source;
constant D ← destination;
P ← S;
while $P \neq D$ **do**
  $f \leftarrow$ face de G contenant P qui coupe PD;
  traverser f encore jusqu'à atteindre le bord UV qui coupe PD à un point $P' \neq P$;
  P ← P';
end

le moins de trafic de contrôle. Elle est donc plus efficace en terme de consommation d'énergie. Cependant, elle nécessite une localisation assez précise des nœuds du réseau.

2.2.4 Protocoles avec gestion de la QdS

2.2.4.1 SAR

SAR [Sohrabi 2000] (pour Sequencial Assignement Routing) est un protocole multi-chemin et utilisant les tables de routage. Il était parmi les premiers protocoles de routage développés pour les réseaux de capteurs sans fil considérant les besoins en termes de QdS. L'objectif de ce protocole est de créer plusieurs arbres dont *les racines sont les nœuds qui sont à un seul saut du puits*. Lors de la construction, chaque arbre évite les nœuds qui ont une faible énergie et qui ne peuvent pas assurer les besoins en termes de QdS (i.e. bas débit/délai important). Ainsi, plusieurs chemins connectent n'importe quel nœud du réseau au puits.

Chaque nœud spécifie deux paramètres pour chaque chemin vers le puits :

- **Ressources énergétiques :** les ressources énergétique d'un chemin sont estimées en calculant le nombre maximal de paquets qui peuvent être transmis par le nœud, s'il est le seul à utiliser ce chemin.
- **Métrique de QoS additive :** chaque chemin a une métrique de QdS additive définie en se basant sur l'énergie et le délai de chaque lien du chemin. Une grande valeur exprime une QdS faible.

Pour effectuer le routage, un nœud choisit le chemin qui correspond le mieux à la QdS exigée par le paquet. Cette exigence est exprimée par la priorité du paquet.

2.2.4.2 Minimum Cost Path Forwarding

MCPF [Ye 2001] utilise le coût des liens pour établir les chemins vers le puits. Le coût d'un lien est calculé en fonction du délai, du débit et de la consommation d'énergie du nœud.

MCPF opère en deux phases : (1) l'établissement des coûts et (2) transmission des données. L'établissement des coûts démarre par un message ADV émis par le puits et contenant un coût nul. Tous les autres nœuds initialisent leur coût à une valeur infinie. Lorsqu'un nœud reçoit un message ADV, il vérifie si la valeur reçue additionnée au coût du lien est plus petite que la valeur locale. Dans ce cas, le nœud met à jour sa valeur locale et émet un nouveau message ADV comme suit ; le nœud choisit un délai de backoff proportionnel à la nouvelle valeur du coût et envoie le message à l'expiration de ce délai. Le délai du backoff permet aux nœuds suivants de mettre à jour leurs coûts en sélectionnant celui qui a le coût minimum vers le puits. La phase de transmission démarre après. Lorsqu'un paquet est émis par une source vers le puits, il contient le coût minimal local du nœud. A la réception d'un paquet de donnée, le nœud vérifie si son coût local est égal au coût reçu moins le coût du lien de réception. Dans ce cas, le nœud relaie le paquet en remplaçant la valeur du coût par sa valeur locale. Sinon, le message est supprimé.

MCPF n'utilise aucune identification des nœuds et aucune table de routage. Ainsi, il consomme peu de ressources mémoire.

2.2.4.3 SPEED

SPEED [He 2003] est un protocole de routage à temps réel souple basé sur la localisation géographique. Il part de l'hypothèse suivante : le délai de bout-en-bout dépend de la distance entre la source et la destination. Il essaie, ensuite, de garantir la vitesse de délivrance des paquets à travers le réseau de capteurs de manière à ce que ce délai de bout-en-bout soit proportionnel à la distance entre la source et la destination. La vitesse de délivrance est une contrainte qu'on peut définir au niveau application.

Le routage dans SPEED se fait de la manière suivante : Lorsqu'un paquet arrive à un nœud i celui-ci sélectionne les voisins qui sont plus proches que lui de la destination. Ensuite parmi cet ensemble, il sélectionne un sous ensemble de nœuds qui ont une vitesse de relais supérieure à une certaine valeur fixée S_{set} (qui dépend de la taille du paquet, de la bande passante disponible et de la couverture radio). La vitesse de relais est le quotient distance entre i et le voisin de i sur le délai pour atteindre ce voisin. Ensuite, le candidat pour la transmission est choisi parmi cette sous liste (le choix est probabiliste). Enfin, si cette sous liste est vide, un premier mécanisme d'adaptation de trafic est appelé pour tenter de diminuer la valeur de S_{set}. Dans le pire des cas le paquet est supprimé.

Afin de maintenir la vitesse désirée, SPEED emploie un deuxième mécanisme de régulation du trafic qui permet de diminuer la charge sur un nœud donné en agissant sur le trafic émis par tous ses prédécesseurs. Le trafic est ou bien détourné vers d'autres nœuds ou bien supprimé.

Plusieurs variantes de SPEED ont été développées comme par exemple "two-hop SPEED" [Li 2009] qui améliore les décisions de routage en utilisant des vitesses de relais à deux-sauts. "Multi-path and Multi-speed Routing" [Felemban 2006],

MMSPEED, est une autre variante de SPEED qui utilise des chemins multiples et supporte la différenciation de service à travers l'utilisation de vitesses de relais qui varient en fonction de la classe du trafic.

2.2.4.4 RPAR : Real-time Power-Aware Routing

RPAR [Chipara 2006] est un protocole de routage dit à temps réel souple : Il essaie de garantir les délais de communication exigés par les applications tout en consommant moins d'énergie. Il se base sur l'hypothèse suivante : plus l'énergie est élevée plus les délais de transmission sont faibles. Ainsi le protocole établit un compromis entre la consommation d'énergie et les délais de transmissions.

On appelle *vélocité* le rapport entre la distance parcourue par un paquet et son délai de bout-en-bout.

RPAR se base sur le routage géographique et l'améliore. L'amélioration consiste à adapter dynamiquement la puissance de transmission selon les exigences de l'application. En effet, RPAR considère chaque paquet à part lors du processus de routage. Il transmet le paquet au nœud qui offre le meilleur choix en terme de consommation d'énergie (donc en terme de puissance de transmission). Un nœud voulant transmettre un paquet calcule les vélocités offertes par ses voisins (appelés choix de transmission). Il détermine ensuite ceux qui peuvent offrir une vélocité supérieure à celle requise par le paquet (la vélocité requise pour assurer l'arrivée dans les temps du paquet) ; ce groupe de nœuds est appelé choix de transmissions éligibles. Enfin, il estime le coût d'énergie qui sera consommée lors de la transmission du paquet aux nœuds éligibles. Le prochain nœud choisi sera donc celui qui offre la vélocité requise tout en consommant le moins d'énergie.

2.2.5 Standards

2.2.5.1 ZigBee

ZigBee se base sur le standard IEEE 802.15.4 et spécifie les couches réseaux et applications. Nous nous intéressons uniquement à la couche réseau.

Vue d'ensemble de la couche réseau : ZigBee possède trois types de périphériques :
- Zigbee Coordinator (ZC) : Il agit comme un coordinateur du PAN défini dans IEEE 802.15.4. Un seul ZC est requis dans le réseau.
- Routeur ZigBee (ZR) : Il agit comme un coordinateur défini dans IEEE 802.15.4. Il participe au routage multi sauts et sert principalement à étendre le réseau.
- Périphériques Zigbee (ZED) : Ils assurent les activités requises du réseau comme la détection par exemple. Ils sont limités en ressources et ne participent pas au routage.

La couche réseau inclut des mécanismes pour :

– Création d'un nouveau réseau et l'attribution des adresses (ou bloc d'adresses) aux nouveaux nœuds. Ces fonctions sont assurées uniquement par le ZC.
– Joindre ou quitter un réseau.
– Sécuriser le transfert de données.
– Routage des trames.
– Découverte et maintenance des routes entre les périphériques.
– Découverte des voisins directs.
– Sauvegarde d'informations sur les voisins.

La couche réseau supporte les topologies en étoile, en arbre ainsi que celle maillée. Dans la première, le réseau est contrôlé par le coordinateur ZigBee (ZC) qui, en plus des fonctions de création du réseau et de sa maintenance, assure la communication de tous les ZED entre eux. En effet, tous les autres nœuds ne communiquent qu'avec le coordinateur ZigBee.

Dans les deux autres topologies (en arbre et maillée), le coordinateur ZigBee est responsable de la création du réseau et de la définition de certains paramètres réseau. Le réseau peut, ensuite, être étendu en utilisant des routeurs ZigBee. Les réseaux en arbre peuvent utiliser la structure de supertrame pour la communication (mode BEACON défini dans IEEE 802.15.4) et le routage se fait d'une manière hiérarchique. Le réseau maillé, en revanche, permet une communication pair-à-pair entre les nœuds et utilise donc le mode sans BEACON.

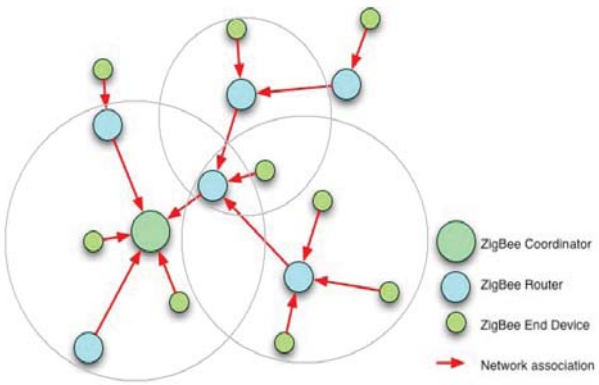


FIGURE 2.14 – Exemple d'un réseau ZigBee.

Structure du réseau et adressage : La figure 2.14 montre une structure typique d'un réseau ZigBee composée de plusieurs clusters. Après la formation du cluster principal par le coordinateur ZigBee, les routeurs ZigBee permettent d'étendre la structure du réseau en ajoutant d'autres associations (flèches rouges).

L'attribution d'adresses dans ZigBee se fait de deux manières selon la valeur de la variable booléenne *nwkUseTreeAddrAlloc*. Dans le cas où elle contient la valeur "False", les adresses sont allouées d'une manière incrémentale par la couche ap-

plication jusqu'à l'épuisement des adresses. Dans l'autre cas, l'adressage suit une structure d'arbre et est fait de façon à allouer équitablement à chaque parent potentiel une partie des adresses qu'il peut distribuer à ses fils. Le coordinateur ZigBee détermine lors de la formation du réseau le nombre maximal de fils par parent ainsi que le nombre maximal de routeurs parmi ces fils. De plus, à chaque nœud, est associé une valeur appelée "depth" (profondeur) qui est le nombre minimal de sauts pour atteindre le coordinateur ZigBee. Ainsi, en considérant le nombre maximal de fils qu'un parent peut avoir (Cm), la profondeur maximale du réseau (Lm) et le nombre maximal de routeurs parmi ces fils (Rm), on calcule la fonction $Cskip(d)$(équation 2.5) qui représente la taille du sous-bloc d'adresses pouvant être allouées à un parent se trouvant à la profondeur d

$$Cskip(d) = \begin{cases} 1 + Cm \cdot (Lm - d - 1), & \text{si Rm=1} \\ \frac{1+Cm-Rm-Cm \cdot Rm^{Lm-d-1}}{1-Rm}, & \text{sinon} \end{cases} \tag{2.5}$$

La distribution des adresses dans la structure d'arbre se fait comme suit (voir figure 2.15) :

- Le coordinateur ZigBee possède l'adresse 0.
- Si le nœud est un routeur ZigBee (ZR) alors :
  - S'il est le premier servi : Adresse = Aparent+1.
  - Sinon : $Adresse = A_{attribuée\ auparavant} + Cskip(d)$.
- Si le nœud est un simple périphérique (ZED) : L'adresse d'un nœud n, se trouvant à la profondeur $d+1$ serait

$$A_n = A_{parent} + Cskip(d) \cdot Rm + n \tag{2.6}$$

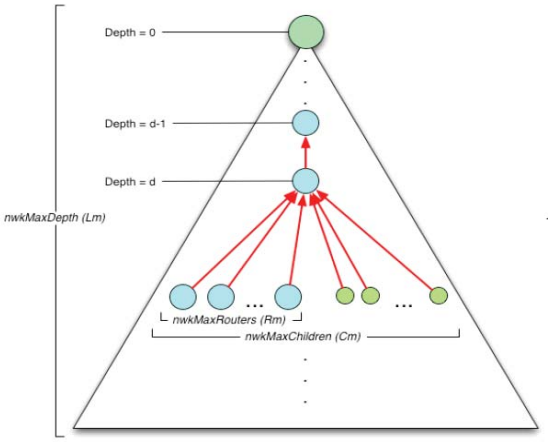


FIGURE 2.15 – Distribution d'adresses dans la structure d'arbre.

Routage : La règle par défaut pour router une trame dans ZigBee est la suivante : si le nœud dispose d'une table de routage et il existe une entrée dans la table alors l'utiliser. Sinon, s'il n'existe pas une entrée dans la table de routage et il existe une place libre alors lancer une requête "RouteDiscovery" (découverte de route). Sinon, router le long de l'arbre en utilisant le routage hiérarchique (RA). Dans le dernier cas, le routage hiérarchique ainsi que l'adressage selon la structure d'arbre doivent être activés.

La découverte de route utilise une version légèrement modifiée de AODV [11]. En effet, le processus de découverte est exactement le même qu'AODV (diffusion de la requête vers tout le réseau et la destination ou les nœuds connaissant un chemin vers la destination répondent). La seule différence se situe dans la fonction de coût d'un lien. Le coût d'un lien 'l' est donné par l'équation suivante :

$$C\{l\} = min(7, round(\frac{1}{p_l^4})) \tag{2.7}$$

où p_l est la probabilité de délivrance de paquets sur le lien l et $round(x)$ est la fonction d'arrondi. Le routage hiérarchique se base sur quelques règles de comparaison et tire profit de l'adressage selon la structure d'arbre. Le fonctionnement est comme suit :

Considérons une trame qui a pour destination 'D' et qui arrive au niveau du routeur ZigBee 'A' ayant comme profondeur 'd'. Si l'expression suivante est vraie alors D est un de ses descendants :

$$A < D < A + Cskip(d-1) \tag{2.8}$$

Dans ce cas, la règle suivante permet de déterminer le saut suivant (N) :

– $N = D$ pour les simples périphériques (ZED) tel que $D > A + Rm \cdot Cskip(d)$
– sinon :

$$N = A + 1 + \left\lfloor \frac{D - (A+1)}{Cskip(d)} \right\rfloor \times Cskip(d) \tag{2.9}$$

Si l'expression 2.8 n'est pas vraie alors envoyer le paquet au père.

Une première amélioration a été proposée afin d'optimiser le délai de transmission de bout-en-bout d'un paquet. Le paragraphe suivant détaille cette amélioration.

Routage en Arbre Amélioré : Considérons une trame qui a pour destination 'D' et qui arrive au niveau du routeur ZigBee 'A' ayant comme profondeur 'd'. Nous désignons par $V(x)$ la liste des voisins du nœud x et $d(x)$ la profondeur du nœud x. L'ensemble des règles du routage modifié (nous le désignerons par *R2A* pour Routage en Arbre Amélioré par la suite) est le suivant :

11. Ad-Hoc On Demand Distance Vector, c'est un protocole de routage ad-Hoc [Perkins 2003]

1. **Si** D est un descendant du nœud A, **alors** Utiliser la règle donnée par l'équation 2.9 pour trouver le saut suivant.
2. **Sinon**, **Pour** tout N dans $V(A)$
   Si D est un descendant de N (Utilisant la règle donnée par l'équation 2.8), **alors** $Saut\text{-}suivant = N$.
3. **Sinon** (n'est pas un descendant, n'est pas un voisin et n'est le descendant d'aucun voisin de A) Envoyer le paquet au père.

S'il y a plusieurs voisins qui peuvent atteindre la destination, choisir celui qui a la profondeur la plus grande.

La principale modification est située au niveau de la deuxième étape de l'algorithme. R2A utilise la table des voisins pour améliorer ses décisions de routage. Cet algorithme améliore celui de base sans toutefois le dégrader car le pire cas dans R2A correspond à la route établie par l'algorithme de base.

Exemple : Considérons un réseau composé de 10 nœuds répartis comme le montre la figure 2.16.

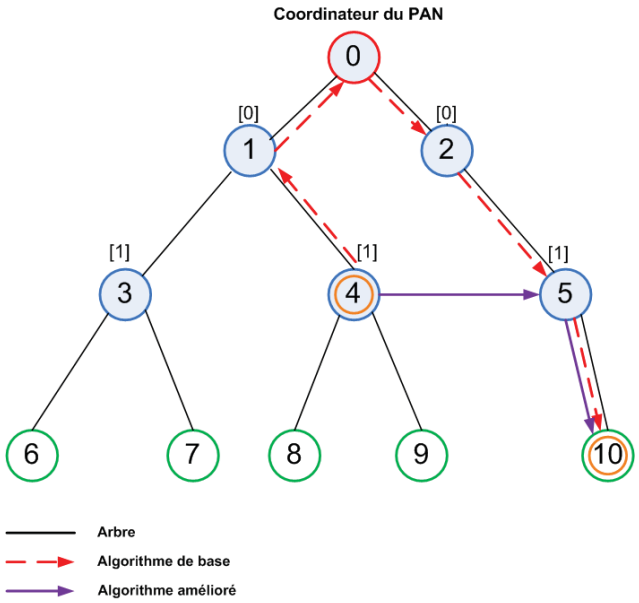


FIGURE 2.16 – Routage hiérarchique et routage hiérarchique amélioré.

Supposons que le nœud 4 envoie des données au nœud 10 et analysons les chemins pris par les deux algorithmes de routage hiérarchique.

La construction du chemin pour le RA se fait comme suit :

– Le nœud 10 n'est pas un descendant du nœud 4. Le message est envoyé vers le père qui est le nœud 1.

– Le nœud 10 n'est pas un descendant du nœud 1 donc celui-ci envoie le message à son père (nœud 0).
– Le nœud 4 est un descendant du nœud 0. Celui-ci détermine le saut suivant qui est dans notre cas le nœud 2.
– Le nœud 4 est un descendant du nœud 2. Celui-ci envoie le message vers le fils approprié (il y a qu'un seul) le nœud 5.
– Le nœud 5 envoie directement le message à son fils.

Le nombre de sauts est égal à 5 (ligne rouge dans la figure 2.16)

La construction du chemin pour le R2A se fait comme suit :

– Le nœud 10 n'est pas un descendant du nœud 4.
– Le nœud 4 vérifie si le nœud 10 appartient à la descendance d'un de ses voisins. Il trouve que le nœud 10 est un descendant de deux voisins : les nœuds 2 et 5. Il choisit donc le nœud 5 puisqu'il a la profondeur la plus grande.
– Le message est envoyé vers le nœud 5. Celui-ci livre le message directement au nœud 10.

Le nombre de sauts est égal à 2 (ligne en violet dans la figure 2.16).

2.2.5.2 ROLL

ROLL ("Routing Over Low power and Lossy networks") est un groupe de travail formé par l'IETF pour proposer et standardiser un protocole de routage destiné aux applications embarquées opérant sur des réseaux composés de nœuds ayant des contraintes d'énergie, de mémoire et de capacité de traitement. ROLL utilise le terme "Low power and Lossy networks" (LLNs) pour définir de tels réseaux. Les nœuds sont interconnectés en utilisant plusieurs technologies de liaison de données, comme par exemple IEEE 802.15.4, Bluetooth, Wifi, technologies filaires, *etc.*

Le protocole proposé par ROLL, se nomme RPL ("routing protocol for Low Power and Lossy networks"). Par la suite nous décrivons ses principes fondamentaux en se basant sur le dernier draft Internet [IETF 2010].

Organisation et construction de la topologie : La topologie est organisée selon un graphe acyclique et dirigé, DAG ("Directed Acyclic Graph"), lui même divisé en un ou plusieurs DAG orienté(s) vers la ou les destination(s), DODAG ("Destination Oriented DAG). Un DODAG est un DAG qui a une seule racine, la destination. Si un DAG a plusieurs racines, il est prévu que ces racines sont connectées par un backbone commun.

RPL utilise quatre identificateurs pour identifier et maintenir une topologie :

- **RPLInstanceID :** identifie un ensemble de DODAGs. Le réseau peut avoir plusieurs RPLInstanceID, chacune identifiant un ensemble indépendant de DODAGs pouvant être optimisés pour des métriques et/ou des applications particulières.
- **DODAGID :** identifie un DODAG dans une instance RPL.

- **DODAGVersionNumber :** identifie la version actuelle du DODAG. La racine d'un DODAG incrémente cette valeur pour initier une reconstruction du DODAG.
- **Rank :** définit la position des nœuds dans un DODAG par rapport à la racine du DODAG. Elle établit donc un ordre partiel sur le DODAG.

Une instance de RPL peut être composée d'un seul DODAG avec une seule racine, de plusieurs DODAG avec des racines indépendantes, d'un seul DODAG avec une racine virtuelle composée de plusieurs racines connectées sur un seul backbone ou d'une combinaison de ces scénarios.

Notons que chaque DODAG construit n'a pas forcément une structure d'arbre puisque chaque nœud peut avoir plusieurs parents à la fois. Ainsi, la structure en arbre n'est qu'un exemple particulier. La figure 2.17 donne un exemple d'un réseau LLN composé de deux DODAGs dont les deux racines sont connectées sur un même backbone. Par ailleurs, la construction du DODAG se fait par rapport à une fonction objectif (OF) qui détermine comment les nœuds sélectionnent et optimisent les chemins vers la racine. L'OF définit comment les nœuds traduisent une ou plusieurs métriques et contraintes vers une valeur appelée Rank. Elle définit aussi comment les nœuds sélectionnent leurs parents. La construction d'un DODAG se fait comme suit :

1. Quelques nœuds sont configurés pour devenir des racines
2. Les nœuds annoncent leur présence, appartenance à un DODAG, coût de routage et les métriques en envoyant des messages DIO (DODAG Information Object) en multicast.
3. Les nœuds reçoivent les DIO et utilisent les informations qui sont dedans pour joindre un nouveau DODAG et sélectionner leurs parents ou pour maintenir un DODAG existant.
4. Les nœuds mettent à jour leurs tables de routage pour les destinations spécifiées dans le message DIO. Notons que chaque destination, hors la racine, doit annoncer sa présence à la racine pour permettre la construction de chemins vers elle. Ceci se fait à travers des messages DAO (Destination Advertisement Object). Les messages DIO contient donc la liste des destinations connues au moment de la création du message.

Notons donc que RPL ne crée de routes qui ont des portions descendantes (dans le sens racine-réseau) que pour les destinations qui ont annoncé leur présence. Cela dit, une destination donnée, peut créer un DAG, dit local, permettant de router les messages vers elle.

Transmission des données et routage : RPL supporte trois types de trafic : multipoint-à-point (MPàP), point-à-multipoint (PàMP) et point-à-point (PàP). Le trafic MPàP se passe entre les nœuds du réseau et la racine. Le routage se fait à travers le graphe construit. Un nœud peut sélectionner un ou plusieurs pères pour relayer le message, en fonction de la métrique à optimiser. Le trafic PàMP se passe

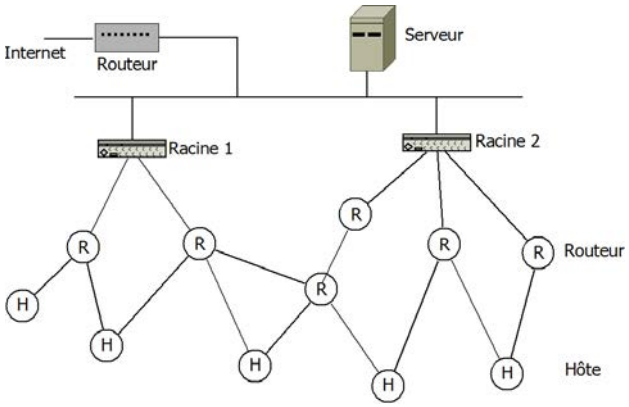


FIGURE 2.17 – Exemple d'un réseau LLN composé de deux DODAGs dont les deux racines sont connectées sur un même backbone. Quelques nœuds appartiennent simultanément aux deux DODAGs.

généralement entre la racine et plusieurs nœuds du réseau. Chaque destination doit annoncer sa présence en envoyant un message DAO à la racine à travers ces pères. Les nœuds intermédiaires mettent à jour leurs tables de routage avant de relayer le ou les messages. Le trafic PàP se passe entre deux nœuds quelconques du réseau. A chaque pas, le routeur transmet le message à un père si la destination n'est pas un de ses descendants. Ce mécanisme constitue le protocole de routage par défaut. RPL autorise l'utilisation d'autres protocoles de routages.

2.2.6 Discussion

La figure 2.18 présente une classification des protocoles étudiés selon leur nature. Notons que cette étude n'inclut pas les protocoles développés pour les réseaux ad-hoc sans fil et qui sont utilisés aussi pour les réseaux de capteurs sans fil, tel que AODV et DSR [Johnson 2007].

Les protocoles de routage doivent satisfaire plusieurs contraintes à la fois telles que l'optimisation de la mémoire utilisée, l'optimisation des traitements de données, l'énergie, la densité du réseau et plus récemment satisfaire plusieurs critères de QdS. Dans la pratique, aucun protocole de routage développé ne présente une solution unique pour toutes ces problématiques. Par ailleurs, le protocole MAC influe considérablement sur la performance des protocoles de routage. Par exemple, la plupart des protocoles étudiés ne prennent pas en compte l'impact de l'endormissement des capteurs [YANG 2011]. Par exemple, les protocoles qui nécessitent un échange de messages pour l'élaboration des routes, comme DD et SPIN, sont beaucoup plus influencés par le mécanisme d'endormissement des nœuds. L'élaboration des routes nécessite un temps beaucoup plus long dans ce cas. De plus, le réseau peut devenir très vite congestionné lorsque la durée de la partie active n'est pas assez grande. Ainsi les choix que nous devons faire lors de la conception de la couche MAC doivent

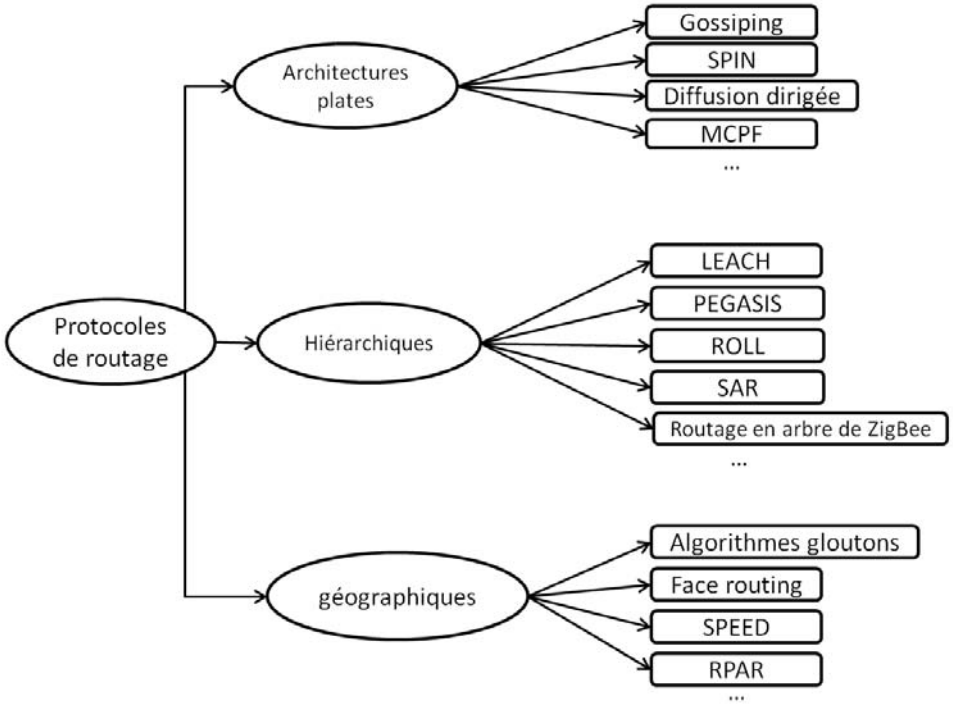


FIGURE 2.18 – Classification des protocoles de routage étudiés.

tenir compte des caractéristiques du protocole de routage à utiliser. Dans ce travail nous retenons l'utilisation d'un protocole de routage hiérarchique, en l'occurrence, le routage en arbre de ZigBee. Ce choix est dû à plusieurs raisons. D'abord, le protocole est simple et son implémentation est triviale. Ensuite, il ne nécessite aucun échange de paquets pour l'établissement des chemins. Par conséquent, Il est possible de l'utiliser sur n'importe quelle couche MAC sans que celle-ci altère son fonctionnement d'une manière significative. Puis, à l'inverse du routage géographique, il n'a besoin d'aucun dispositif ou protocole de localisation. Enfin, la plupart des réseaux de capteurs sans fil envoient les données vers un nombre limité de destinations. Ainsi, le routage hiérarchique s'impose comme un choix naturel pour relayer les messages. Certes, le routage en arbre constitue un bon choix pour le routage dans les réseaux de capteurs sans fil, mais il a aussi plusieurs inconvénients qui doivent êtres traités afin d'offrir un protocole de routage performant. En effet, il n'est pas assez robuste face aux changements dynamiques du réseau ; la mort d'un nœud peut nécessiter par exemple la reconfiguration de toute la branche des descendants, ce qui induit un temps long durant lequel les nœuds sont incapables de reporter les données captées. Par conséquent, le protocole de base est peu fiable. De plus, le nombre de sauts n'est généralement pas optimal et ne prends pas en compte l'état du prochain saut tel

que son énergie restante, sa charge, *etc.* Finalement, nous retenons l'utilisation du routage hiérarchique et l'améliorons.

2.3 Conclusion

Dans ce chapitre nous avons présenté un état de l'art sur les protocoles MAC et routage dans les réseaux de capteurs sans fil. Pour les protocoles MAC nous avons remarqué que la plupart des solutions ont pour objectif l'optimisation de la consommation d'énergie. On observe aussi que les protocoles gérant la QdS se basent généralement sur les protocoles CSMA/CA, TDMA ou une combinaison des deux. Constatant les faiblesses et les forces de chaque solution, notre objectif est de proposer un protocole MAC qui assure une gestion de la QdS avec le coût le plus faible possible. Le coût peut être défini en fonction de celui de fabrication des nœuds, de déploiement du réseau, de la durée de vie du réseau, *etc.* Pour les protocoles de routage, nous avons conclu que les choix que nous devons faire lors de la conception de la couche MAC doivent tenir compte des caractéristiques du protocole de routage à utiliser. L'inverse est vrai aussi ; la conception d'un protocole de routage doit prendre en compte la couche MAC à utiliser. Ainsi nous pouvons constater que la conception et l'optimisation conjointe des deux couches MAC et réseau est nécessaire afin de répondre aux exigences des réseaux de capteurs sans fil en général et proposer un mécanisme de gestion de QdS efficace et peu coûteux. Sachant que la QdS est la responsabilité de toutes les couches protocolaires, nous pouvons pousser le raisonnement plus loin et dire qu'une optimisation conjointe de toutes les couches est la meilleure solution pour la gestion de la QdS dans les réseaux de capteurs sans fil.

Dans cette thèse nous avons choisi comme protocoles de base CSMA/CA et le RA (et sa version améliorée R2A) au niveau de la couche MAC et de la couche réseau, respectivement. Nous proposons ainsi un ensemble de solutions basés sur ces choix avec une optimisation conjointe afin de supporter la QdS. Le choix du RA a été argumenté dans la sous-section 2.2.6. les motivations du choix de CSMA/CA seront détaillées au début du chapitre suivant qui présente CoSenS. L'objectif de CoSenS est d'améliorer les performances du protocole d'accès CSMA/CA et de faciliter l'implémentation de mécanismes de QdS.

Chapitre 3

CoSenS - Collecter puis Envoyer en burst

Sommaire

L'objectif principal de ce travail de thèse consiste à fournir un ensemble de solutions permettant l'offre de la QdS dans les réseaux de capteurs sans fil. Nous présentons dans ce chapitre notre première approche, que nous avons appelée CoSenS, développée pour la couche MAC. CoSenS constitue le pilier de base de cet ensemble.

Le chapitre est organisé comme suit. Dans la première section, nous décrivons les motivations qui nous ont amené à proposer CoSenS. Dans la section 3.2, nous détaillons son fonctionnement. Les sections 3.3 et 3.4 présentent l'évaluation de performance de CoSenS par simulation et par implémentation sur la plate-forme de capteurs Jennic [Jennic Ltd. 2011c]. La section 3.5 conclut le chapitre.

3.1 Motivations

Nous avons remarqué dans le chapitre précédent que les solutions d'accès avec gestion de la QdS sont ordonnancées, basées sur la contention ou hybrides. La première question à laquelle nous devons répondre consiste à déterminer le type de protocole d'accès à utiliser afin de résoudre la problématique de la QdS au regard de l'idée que nous soutenons, à savoir une solution de gestion de la QdS best-effort, simple, peu coûteuse, suffisamment flexible et pouvant s'auto-adapter aux évolutions du réseau.

3.1.1 Améliorer la performance à faible coût

La plupart des solutions ordonnancées et hybrides proposées dans la sous-couche MAC pour le support de la QdS se basent sur la technique TDMA. Cette technologie a l'avantage de fournir des performances déterministes qui sont en plus très bonnes dans les cas où la charge sur le réseau est importante, cas où la notion de QdS prend tout son sens. Cependant, cette technique présente plusieurs inconvénients, rendant son utilisation difficile et restreinte dans les réseaux de capteurs sans fil. Premièrement, elle nécessite une configuration minutieuse du réseau afin d'ordonnancer les slots de transmission de tous les nœuds. Deuxièmement, tous les nœuds doivent être régulièrement synchronisés avec une grande précision. Ainsi, du point de vue matériel, un nœud qui utilise cette technique doit embarquer une horloge temps réel utilisant un cristal de haute qualité pour limiter la dérive de l'horloge et un processeur puissant pour supporter les algorithmes de synchronisation. Troisièmement, TDMA s'adapte difficilement aux différents changements du réseau. Or les réseaux de capteurs sans fil sont très dynamiques en raison des changements fréquents de l'état du canal sans fil, de la mobilité, de l'épuisement de la batterie, de l'environnement de déploiement, *etc.*

3.1.2 Choix technologiques et idée de base

Afin de réaliser notre objectif, nous avons opté pour l'utilisation du protocole CSMA/CA. En effet, il est très simple à implémenter et ne nécessite ni une synchronisation entre les nœuds, ni un effort particulier de configuration. Ce protocole offre de bonnes performances dans le cas où la charge sur le réseau est faible, ce qui correspond à un usage normal du réseau. Cependant, elles se dégradent au fur et à mesure que la charge augmente (faibles taux de succès de transmission, délais de bout-en-bout importants et faibles débits), ce qui correspond à des périodes de surcharge du réseau (burst). Les périodes de surcharge sont inévitables et surviennent souvent suite à la détection d'un problème dans l'environnement surveillé impliquant un échange important de données. CSMA/CA tel quel est considéré inadapté au support de la QdS dans la couche MAC.

Nous proposons donc dans un premier temps d'améliorer les performances intrinsèques de CSMA/CA. Pour cela, nous avons développé une nouvelle méthode d'accès

que nous avons appelé CoSenS, pour "Collecting then Sending burst Scheme". CoSenS ne modifie pas le protocole CSMA/CA lui-même, qui reste le protocole de base pour accéder au canal, il contrôle plutôt les instants de transmission et les appels à ce protocole. L'idée de base de CoSenS est qu'un nœud ne retransmet pas les paquets dès leur arrivée, il attend plutôt pendant un certain temps et collecte un ensemble de paquets. Après, il transmet le tout dans un seul burst, comme s'il y avait un seul paquet à transmettre (en vérité chaque paquet est acquitté avant de passer au suivant). CSMA/CA n'est donc utilisé qu'une seule fois pour accéder au canal de transmission (s'il n'y a pas d'erreurs de transmission d'un paquet, voir (3.2.4) pour la spécification complète du mécanisme de transmission).

Par la suite, nous développons CoSenS en détail.

3.2 CoSenS en détail

3.2.1 Hypothèses

Nous considérons un réseau organisé selon une architecture deux-tiers. Il est composé de routeurs auxquels sont attachés des nœuds simples (typiquement des FFD et des RFD selon la nomenclature du standard IEEE 802.15.4 (section 2.1.5.1)). Un nœud simple est un dispositif de fin limité en ressources processeur et mémoire. Sa principale fonction est de prélever les mesures de ses capteurs intégrés et les transmettre à un routeur (son père par exemple). Un routeur est un dispositif plus puissant qui implémente tous les protocoles permettant de créer et de gérer un réseau et de relayer des messages vers n'importe quelle destination. Un routeur peut aussi intégrer des capteurs et ainsi agir comme un nœud simple.

Un nœud simple s'associe toujours à un routeur. Une association est une relation de type père-fils où le fils est associé au père. Un nœud simple ne peut pas être père. Un routeur peut être à la fois fils et père. Un routeur peut avoir plusieurs fils. Chaque nœud simple a un et un seul père. Les routeurs peuvent s'organiser selon une architecture plate ou hiérarchique. Les routeurs peuvent utiliser n'importe quel protocole de routage. La figure 3.1 illustre un exemple d'architecture possible où les nœuds simples sont associés à des routeurs qui forment un réseau ad-hoc (voir Annexe A). Si un nœud simple veut transmettre des données, il les transmet à son père. Cette contrainte peut être assouplie en transmettant les données à n'importe quel **routeur** voisin. Toutes les requêtes comme par exemple l'obtention d'information à propos d'une destination (son adresse par exemple) doivent être adressées au père.

Le protocole de base pour accéder au canal de transmission est CSMA/CA non slotté proposé par le standard IEEE 802.15.4. Cependant, les valeurs des paramètres du protocole diffèrent selon le type de nœud ; routeur ou nœud simple. CoSenS est donc implémenté au dessus de CSMA/CA. Par défaut il est activé pour les routeurs et désactivé pour les nœuds simples (voir la sous-section suivante pour la motivation

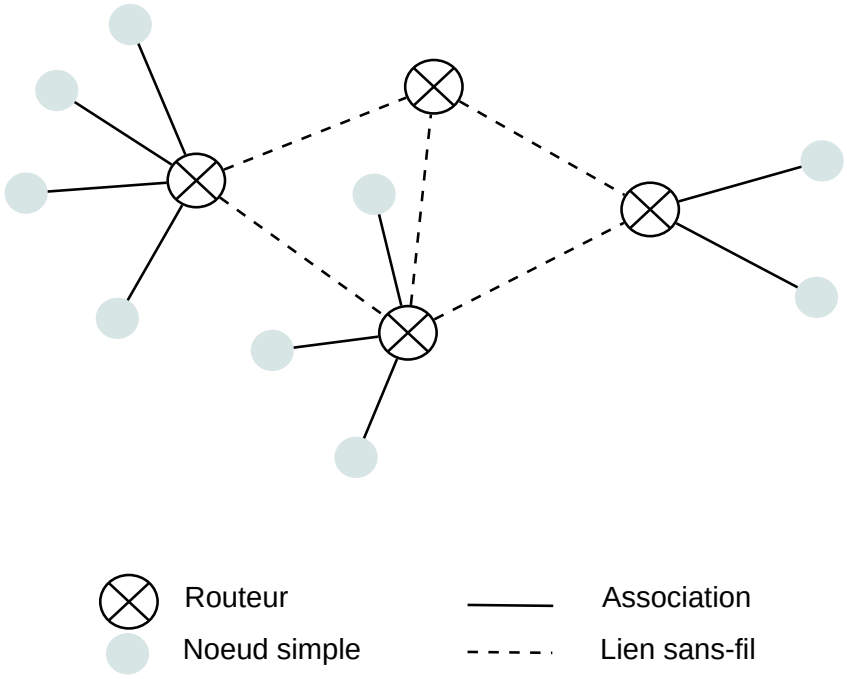


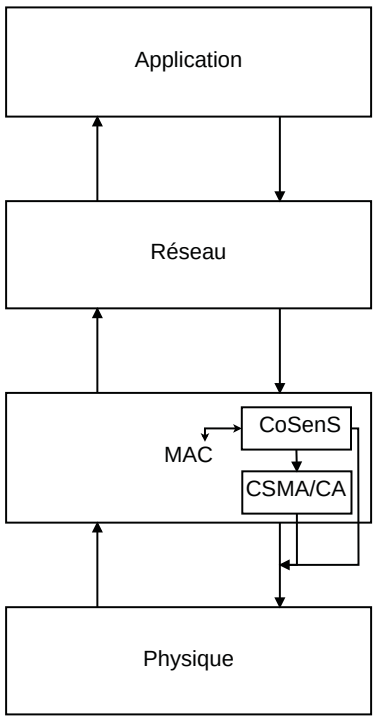
FIGURE 3.1 – Exemple d'architecture possible pour un réseau de capteur sans fil.

de ce choix). Ainsi, dans le reste du chapitre nous utiliserons le terme routeur pour décrire le fonctionnement de CoSenS.

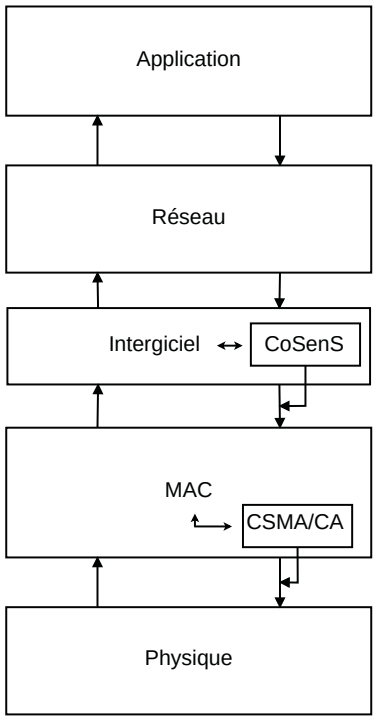
La couche physique utilisée est celle de IEEE 802.15.4 qui opère dans la bande 2.4 GHz.

3.2.2 Pile protocolaire intégrant CoSenS et règles de base

La figure 3.2 illustre la pile protocolaire d'un nœud intégrant CoSenS. Deux choix sont possibles ; CoSenS est ou bien intégré dans la sous-couche MAC (Fig. 3.2(a)) ou bien implémenté dans une couche à part se situant entre la couche MAC et la couche réseau (Fig. 3.2(b)). Dans le premier cas, CoSenS est implémenté directement au dessus du protocole d'accès CSMA/CA. Il implémente les fonctionnalités de récupération des paquets envoyés par la couche réseau et d'arbitrage de l'utilisation de CSMA/CA. Cependant, pour des raisons de compatibilité d'une part et de présence de produits où l'implémentation de la couche MAC est inaccessible d'une autre part, CoSenS peut être implémenté dans une couche intergicielle entre la couche MAC et la couche réseau. CoSenS agit sur le flux de données entre la couche réseau et la couche MAC. Il implémente alors les fonctionnalités de récupération des paquets envoyés par la couche réseau, d'envoie de requêtes de configuration vers la couche MAC, tel que l'activation ou la désactivation de CSMA/CA, et de transmission des paquets de données vers la couche MAC selon le cycle imposé par CoSenS (voir paragraphe suivant).



(a) CoSenS intégré dans la couche MAC.



(b) CoSenS inmplémenté dans une couche intergicielle.

FIGURE 3.2 – Piles protocolaires possibles intégrant CoSenS. Les flèches représentent les interactions entre les différents composants.

CoSenS possède quatre règles de base. Premièrement, les routeurs sont plus prioritaires lors de l'accès au canal de transmission. Ceci est réalisé en sélectionnant une valeur plus petite de l'exposant du backoff (un paramètre de CSMA/CA). Deuxièmement, un routeur ne transmet pas les paquets un par un dès leurs arrivés mais il attend pendant une période de temps, appelée période d'attente ou WP [1], et rassemble les paquets qui arrivent de la couche réseau. Ces paquets peuvent provenir de la couche application du routeur lui-même, d'un de ses fils ou d'un des routeurs voisins. Troisièmement, à la fin de la période d'attente, le routeur commence à transmettre tous les paquets collectés dans un seul burst. Cette période est appelée période de transmission ou TP [2]. Finalement, à la fin de la période de transmission, le routeur passe à nouveau en mode attente pour collecter les paquets. Ainsi, CoSenS opère d'une manière cyclique, où un cycle est composé d'une période d'attente et d'une période de transmission (Fig. 3.3). Si aucun paquet n'est collecté durant la période d'attente, la durée de TP est nulle.

Les nœuds simples, transmettent les données pendant les WP de leurs pères puisqu'ils utilisent CSMA/CA. En effet, CSMA/CA vérifie l'état du canal avant de

1. WP : "Waiting Period" en anglais
2. TP : "Transmission Period" en anglais

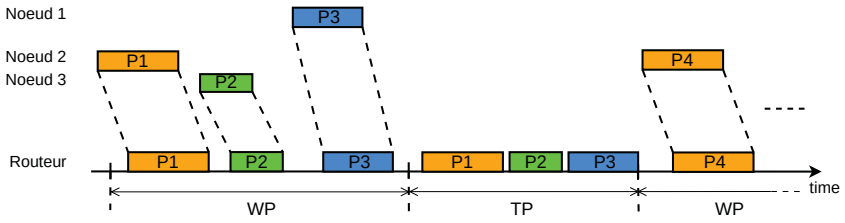


FIGURE 3.3 – Un exemple d'opération de CoSenS.

transmettre. Celui-ci ne peut être libre que pendant la WP. Les nœuds simples ne collectent pas les données hormis les mesures relevées par leurs couches applications. Généralement, ces mesures ont une période de génération longue. Ainsi, il n'est pas judicieux d'utiliser ici la technique de collecte puis d'envoi en burst, *i.e* CoSenS, car il s'agit souvent de transmettre un seul paquet par période.

Ces règles de base seront détaillées dans les paragraphes suivants.

3.2.3 Priorité d'accès au canal

La priorité accordée aux routeurs pour l'accès au canal de transmission se traduit par une bande passante plus grande par rapport aux nœuds simples. Cela motive la première règle parce que le routeur relaie des paquets provenant de sa couche application, de ses fils et des routeurs voisins.

Le protocole CSMA/CA utilise une constante appelée exposant du backoff, $macMinBE$, pour calculer le délai aléatoire à attendre avant de vérifier l'état du canal. Ce délai, appelé période de backoff, d_{BP}, est donné par l'équation 3.1. Dans (3.1), $aUnitBackoffPeriod$ est égale à $0,32$ ms (une constante définie dans IEEE 802.15.4). Il est clair qu'attribuer une valeur plus petite de $macMinBE$ augmente la probabilité d'accès au support. Ainsi, pour les routeurs, $macMinBE$ est fixée à deux alors qu'elle est égale à trois (valeur par défaut dans IEEE 802.15.4) pour les nœuds simples.

$$d_{BP} = Rand\left[0, 2^{macMinBE} - 1\right] \cdot aUnitBackoffPeriod \tag{3.1}$$

3.2.4 Période de transmission

Les paquets collectés pendant la période d'attente, WP, sont transmis en un seul burst. D'abord, le protocole CSMA/CA est utilisé pour transmettre le premier paquet uniquement. Enfin, à la réception d'un acquittement, le paquet suivant est transmis et ainsi de suite jusqu'à ce que la file d'attente se vide. Nous considérons que tous les paquets doivent être acquittés positivement. Si un acquittement n'est pas reçu pour un paquet donné, la transmission en burst s'arrête. Le paquet non acquitté est alors retransmis en utilisant le protocole CSMA/CA. Le routeur reprend la transmission en mode burst lorsque le paquet est transmis avec succès

(acquittement reçu). Si la transmission échoue après plusieurs tentatives, le paquet est supprimé et le routeur passe au paquet suivant si la file n'est pas vide. Dans ce cas, il transmet le nouveau paquet en utilisant CSMA/CA. Ainsi, tant que le routeur a le droit d'émettre et la transmission des paquets se fait avec succès, le mode burst est actif.

La durée de la période de transmission est variable. Elle dépend du nombre de paquets collectés pendant la période d'attente et des retransmissions éventuelles. Notons que tous les paquets collectés pendant la période d'attente précédente seront transmis dans la période de transmission actuelle. De plus, le burst est constitué de paquets qui n'ont pas forcément la même destination.

Lorsque le routeur ne gagne pas l'accès au support que ce soit au début de la TP ou après un échec de transmission, la période d'attente est prolongée pendant un certain temps à l'issue duquel il tente de nouveau de gagner l'accès au support en utilisant CSMA/CA.

L'utilisation de CSMA/CA pour le premier paquet garantit aussi que toutes les périodes de transmissions des routeurs voisins ne se chevauchent pas. En effet, si un routeur ne gagne pas l'accès au canal de transmission au début de sa période de transmission, il prolonge sa période d'attente jusqu'à la fin de période de transmission du routeur ayant gagné cet accès.

3.2.5 Période d'attente

Dans nos premiers travaux nous avons utilisé des périodes d'attente constantes [Nefzi 2008]. Nous avons remarqué qu'en fonction de la charge simulée, les résultats étaient plus ou moins bons. Nous avons conclu que la période d'attente doit être adaptée en fonction de cette charge du trafic. Le premier constat est que plus la charge est importante, plus la durée de la période d'attente doit être longue, et vice-versa. Ainsi nous avons proposé deux algorithmes d'estimation de la durée de la période d'attente en fonction de la charge du réseau. Chronologiquement, le premier algorithme est développé en premier.

L'idée générale des deux algorithmes est que le calcul de la durée de la prochaine période d'attente est réalisé en fonction de la charge estimée de cette période, elle-même déduite à partir des observations des charges précédentes. Le calcul est effectué juste avant le début de la période d'attente. La figure 3.4 montre un deuxième exemple, plus complet, d'opération de CoSenS en utilisant un diagramme spatio-temporel.

3.2.5.1 Premier algorithme d'estimation dit par seuils

On définit

– d_S et d_R comme étant la durée requise par un nœud simple et un routeur, respectivement, pour transmettre un paquet de donnée avec succès et ce dès la première tentative. Il s'agit de la somme des durées nécessaires pour réaliser l'opération d'évitement de collision (d_{BP}), vérifier l'état du canal (t_{CCA}) et détecter qu'il est

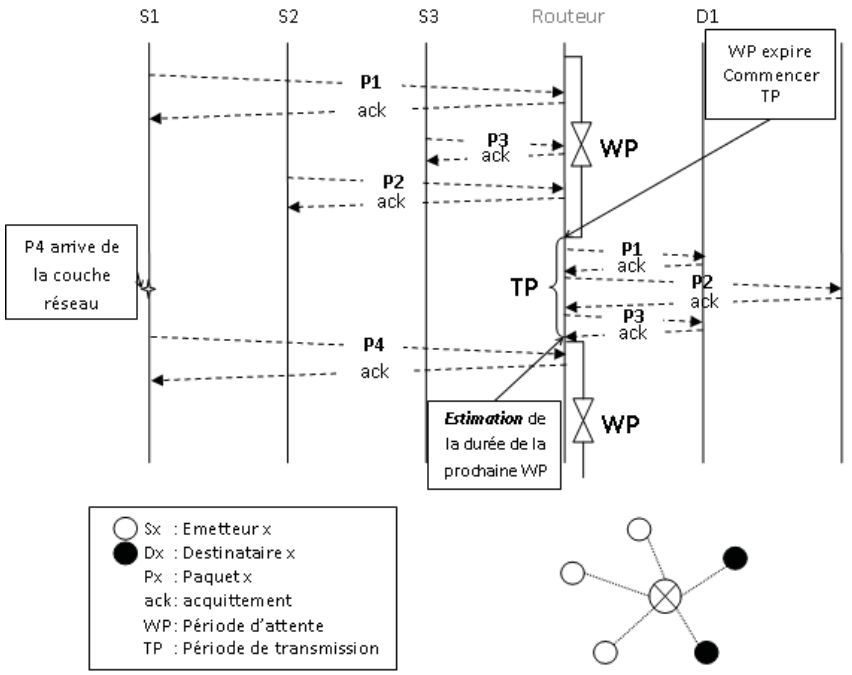


FIGURE 3.4 – Un deuxième exemple d'opération de CoSenS, utilisant un diagramme spatio-temporel.

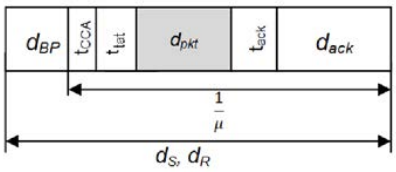


FIGURE 3.5 – Les différents délais qui composent d_S et d_R. $\frac{1}{\mu}$ est le temps de service du paquet.

libre dès la première tentative, changer l'état de l'émetteur-récepteur vers émetteur (t_{tat}), transmettre le paquet (d_{pkt}), attendre la réception de l'acquittement (t_{ack}) et le recevoir (d_{ack}). La figure 3.5 résume les différents délais qui composent d_S et d_R. Remarquons que la somme de t_{CCA}, t_{tat}, d_{pkt}, t_{ack} et d_{ack} est le temps de service du paquet, noté $\frac{1}{\mu}$. La seule différence entre d_S et d_R est la durée maximale de d_{BP}. $d_S > d_R$ du fait que la durée du backoff, d_{BP}, du nœud simple est supérieure à celle du routeur.

– N_d comme étant un entier non nul. Sa valeur dépend de la valeur estimée du trafic.
– N_s^r comme étant le nombre de nœuds simples associés à un routeur r.

La durée d'attente (WP) est donnée par l'équation 3.2. Lorsqu'un routeur n'a aucun nœud simple comme fils, les données proviennent uniquement des routeurs voisins. Ainsi, nous utilisons d_R comme base pour calculer la durée de WP. Dans l'autre cas,

les paquets peuvent arriver des fils et des routeurs voisins. Puisque la valeur d_S est plus grande que d_R, elle est utilisée comme base pour calculer WP, car il faut laisser suffisamment de temps aux nœuds simples pour transmettre leur message. Dans les deux cas, la variable N_d reflète l'estimation de la prochaine charge du réseau. Nous décrivons dans la suite comment nous obtenons cette valeur.

$$WP = \begin{cases} N_d \cdot d_S, & \text{if } N_s^r \neq 0 \\ N_d \cdot d_R, & \text{sinon} \end{cases} \tag{3.2}$$

On définit
- N_s^r comme étant le nombre de nœuds simples attachés au routeur r
- $WP_i,\ i \in \mathbb{N}$ comme étant la $i^{ème}$ période d'attente.
- $\Omega \subset \mathbb{N}$ comme étant l'ensemble des indices, k, des périodes d'attentes pendant lesquelles le routeur reçoit des données.
- $E_k,\ k \in \Omega$, l'esemble des paquets reçus pendant WP_k.
- $tempsService_k = \sum_{i \in E_k} \frac{1}{\mu_i}$ la somme des temps de services des paquets reçus pendant WP_k. $\frac{1}{\mu_i}$ étant le temps de service du paquet i comme le montre la figure 3.5.
- $U_k\ ,\ k \in \Omega$ comme étant le rapport entre $tempsService_k$ et WP_k. U_k est donnée par l'équation 3.3. Remarquons que U_k peut excéder 1 dans le cas où la période d'attente se termine alors que le routeur reçoit toujours un ou plusieurs paquets d'un voisin.

$$U_k = \frac{tempsService_k}{WP_k} \tag{3.3}$$

On note par S_k la moyenne exponentielle glissante de $U_k,\ k \in \Omega$. S_k est donnée par l'équation 3.4

$$S_k = (1 - \alpha)S_{k-1} + \alpha U_{k-1} \tag{3.4}$$

α est le facteur de lissage. Nous avons utilisé un filtre non linéaire où α est plus grand lorsque $U_{k-1} \geq S_{k-1}$. Ainsi, les valeurs plus grandes que la moyenne ont plus de poids. Ceci permet à S_k de s'adapter plus rapidement à l'augmentation brusque du trafic ce qui est le cas lorsqu'un événement survient.

S_k est évaluée juste avant le début de la $k^{ème}$ période d'attente. Après, N_d est calculée en fonction de cette nouvelle valeur de S_k comme décrit dans l'algorithme 3. Dans cet algorithme, U_{MAX}, U_{MIN} et N_{dMAX} désignent, respectivement, le seuil maximal de S_k, le seuil minimal de S_k et la valeur maximale de N_d. U_{MAX} est le seuil d'utilisation maximal au dessus duquel la performance de CoSenS peut s'améliorer en augmentant la durée de la période d'attente. U_{MIN} est le seuil d'utilisation minimal en dessous duquel les performances de CoSenS se dégradent, et donc la durée de la période d'attente doit diminuer. N_{dMAX} limite la valeur de N_d pour éviter des périodes d'attente excessivement longues. Cette valeur peut dépendre de plusieurs facteurs comme par exemple la taille de la file d'attente (plus on attend,

plus la file se remplit). Ce premier algorithme d'estimation est nommé "algorithme d'estimation par seuils".

Algorithm 3: Algorithme d'estimation par seuils.

`/* Mettre à jour` S_k`*/`
$\alpha = \alpha_1$
si $U_{k-1} \geq S_{k-1}$ **alors**
  $\alpha = \alpha_2$
  `/*` $\alpha_2 > \alpha_1$`*/`
fin

$S_k = (1 - \alpha)S_{k-1} + \alpha U_{k-1}$

`/* Mettre à jour` N_d`*/`

si $(S_k \geq U_{MAX})$ **alors**
  $N_d = N_d + 1$
sinon
  si $S_k \leq U_{MIN}$ **alors**
    $N_d = N_d$ -1
  fin
fin

$N_d = \max(1, \min(N_d, N_{dMAX}))$

`/* Mettre à jour la WP */`

si $N_s^r \neq 0$ **alors**
  $WP = N_d \cdot d_S$
sinon
  $WP = N_d \cdot d_R$
fin

S_k calcule le taux moyen d'utilisation de WP. Si S_k dépasse U_{MAX}, la valeur de N_d augmente, ce qui augmente la durée de la période d'attente. Si S_k baisse en dessous de U_{MIN}, N_d diminue, ce qui diminue la durée de la période d'attente. Changer la durée de la période d'attente affecte la valeur de U_k qui, à son tour, influe sur la valeur de S_k. Sachant que U_k (et donc S_k) dépend aussi de la charge du réseau (somme des temps de service), l'algorithme proposé agit comme un processus itératif où l'objectif est de trouver la bonne valeur de N_d (et donc la durée de WP) qui reflète l'intensité du trafic reçu tout en conservant un taux d'utilisation compris

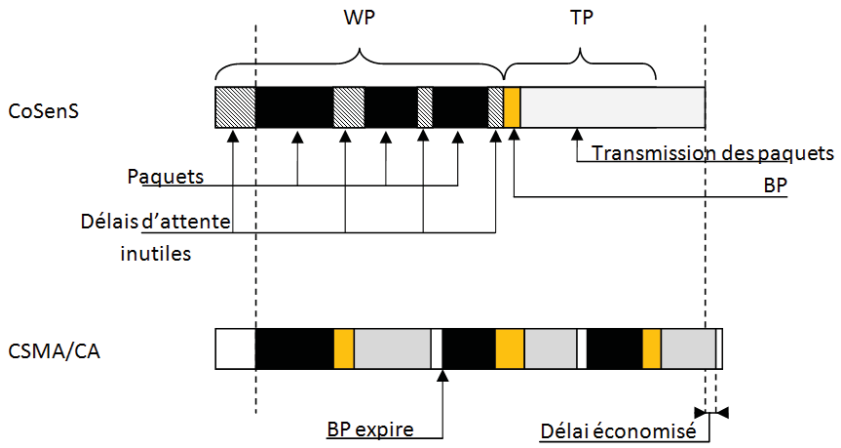


FIGURE 3.6 – Comparaison entre CoSenS et CSMA/CA. La durée BP (représentée en jaune) correspond à la période de backoff de CSMA/CA

entre $[U_{MIN}, U_{MAX}]$.

3.2.5.2 Deuxième algorithme d'estimation dit par comparaison

Le deuxième algorithme répond à la question suivante ; quelle est la meilleure durée de la période d'attente permettant d'améliorer la performance de CoSenS par rapport à CSMA/CA ? Cet algorithme est nommé "algorithme d'estimation par comparaison".

Considérons l'exemple d'un cycle d'attente et de transmission illustré par la figure 3.6. Intuitivement, CoSenS améliore les performances de CSMA/CA si la somme des délais d'attente inutiles (sans réception de paquets de données) au cours de la période d'attente plus un seul délai de backoff (correspondant à la transmission du premier paquet pendant la période de transmission) est inférieure à la somme des délais de backoff qui seront générés au cours de la transmission des paquets reçus par la couche MAC si CSMA/CA seul est utilisé. Ceci est traduit par l'inégalité 3.5. Ainsi la durée de la période d'attente, WP, doit satisfaire la condition donnée par l'inégalité 3.6. Dans ces inégalités, N_{pkts} désigne le nombre de paquets reçus pendant la période d'attente et $pktSvcT_i$ est le temps de service du $i^{ème}$ paquet reçu. $sumSvcT = \sum_{i=1}^{N_{pkts}} pktSvcT_i$.

$$\sum_{i=1}^{N_{pkts}} BP_i \geq WP - \sum_{i=1}^{N_{pkts}} pktSvcT_i + BP_{TP} \tag{3.5}$$

$$WP \leq \sum_{i=1}^{N_{pkts}-1} BP_i + \sum_{i=1}^{N_{pkts}} pktSvcT_i \tag{3.6}$$

$$= \sum_{i=1}^{N_{pkts}-1} BP_i + sumSvcT \tag{3.7}$$

WP est adaptée en ligne de la manière suivante. D'abord, N_{pkts} et $sumSvcT$ sont estimés au début de chaque cycle en se basant sur les observations antérieures. Enfin, WP est calculée selon l'équation 3.8. $\overline{BP}$ est donnée par l'équation 3.9. Elle représente la moyenne de la période de backoff si la transmission réussit dès la première tentative.

$$\overline{WP} = (\overline{N_{pkts}} - 1) \cdot \overline{BP} + \overline{sumSvcT} \tag{3.8}$$

$$\overline{BP} = 2^{macMinBE-1} \cdot aUnitBackoffPeriod \tag{3.9}$$

On définit $\Omega \subset \mathbb{N}$ comme étant l'ensemble des indices, k, des périodes d'attente pendant lesquelles le routeur reçoit des données et $WP_{données} = \{WP_k,\ k \in \Omega\}$ ces périodes d'attente. Notons par $\overline{sumSvcT}$ et $\overline{N_{pkts}}$ les moyennes exponentielles glissantes de $sumSvcT_k$ et N_{pkts_k}, $k \in \Omega$, respectivement.

$$\overline{sumSvcT}_k = (1-\alpha) \cdot \overline{sumSvcT}_{k-1} + \alpha \cdot sumSvcT_k \tag{3.10}$$

$$\overline{N_{pkts}}_k = (1-\alpha) \cdot \overline{N_{pkts}}_{k-1} + \alpha \cdot N_{pkts_k} \tag{3.11}$$

Comme pour l'algorithme d'estimation par seuils, nous utilisons un filtre non linéaire où α est plus grand si $sumSvcT_k \geq \overline{sumSvcT}_{k-1}$ dans l'équation 3.10 et $N_{pkts_k} \geq \overline{N_{pkts}}_{k-1}$ dans l'équation 3.11. $\overline{N_{pkts}}_k$ et $\overline{sumSvcT}_k$ sont évalués avant le début de la $k^{ème}$ période d'attente et utilisés après pour déterminer la durée de cette période d'attente. L'algorithme d'estimation par comparaison est résumé dans Algorithm 4.

3.3 Evaluation des performances par simulation

3.3.1 Données de simulations et métriques de performance

3.3.1.1 Données de simulations

Nous présentons une étude comparative entre CoSenS et IEEE 802.15.4 utilisant CSMA/CA non slotté. Le simulateur utilisé est OPNET [OPNET Technologies].

Algorithm 4: Algorithme d'estimation par comparaison.

/* mettre à jour $\overline{N_{pkts_k}}$ et $\overline{sumSvcT}_k$*/
/* α est sélectionné selon la $(k-1)^{me}$ valeur */
$\alpha = \alpha_1$
si $sumSvcT_k \geq \overline{sumSvcT}_{k-1}$ **alors**
  $\alpha = \alpha_2$
  /* $\alpha_2 > \alpha_1$ */
fin

$$\overline{sumSvcT}_k = (1-\alpha) \cdot \overline{sumSvcT}_{k-1} + \alpha \cdot sumSvcT_k$$

$\alpha = \alpha_1$
si $N_{pkts_k} \geq \overline{N_{pkts}}_{k-1}$ **alors**
  $\alpha = \alpha_2$
fin

$$\overline{N_{pkts_k}} = (1-\alpha) \cdot \overline{N_{pkts}}_{k-1} + \alpha \cdot N_{pkts_k}$$

/* Mettre à jour la WP */
$\overline{WP}_k = (\overline{N_{pkts_k}} - 1) \cdot \overline{BP} + \overline{sumSvcT}_k$
$WP_k = \max(WP_{min}, \min(\overline{WP_k}, WP_{max}))$

L'annexe B décrit ce simulateur et présente les modèles développés pour réaliser cette étude de performance.

Deux protocoles de routages ont été utilisé au cours de cette étude ; le routage statique et le routage en arbre hiérarchique (RA). Quant au premier, il s'agit de spécifier manuellement les tables de routage avant le démarrage de la simulation. Le routage en arbre hiérarchique est celui défini par le standard ZigBee dans la couche réseau (section 2.2.5.1).

Le tableau 3.1 résume les paramètres généraux de simulation. Les paramètres de CSMA/CA sont les mêmes pour CoSenS et IEEE 802.15.4. Sans perte de généralité, la taille du paquet est constante. U_{MIN}, U_{MAX} et les valeurs de α sont déterminés empiriquement. $N_{d_{MAX}}$, WP_{min} et WP_{max} sont fixés arbitrairement. Cependant $N_{d_{MAX}}$ et WP_{max} peuvent être déterminés en fonction du dispositif réel. Leurs valeurs reflètent la capacité du nœud en termes de mémoire. Nous considérons deux types de trafic ; Poisson et périodique. Nous avons simulé soit l'un, soit l'autre, soit les deux. Cinq simulations ont été réalisées pour chaque scénario. C'est la moyenne de ces résultats qui est présentée

Table 3.1 – Paramètres généraux de simulation.

Variable			Valeur
PHY	débit binaire		250 kb/s
Taille paquet	APL		402 bits
	MAC		500 bits
CSMA/CA	$macMinBE$	nœud simple	3
		routeur	2
CoSenS	α	α_1	0.008
		α_2	0.01
	Algorithme d'estimation par seuils		
	U_{MIN}		0.28
	U_{MAX}		0.75
	$N_{d_{MAX}}$		15
	Algorithme d'estimation par comparaison		
	WP	WP_{min}	0.001 s
		WP_{max}	0.07 s
Type de trafic			Poisson/périodique

3.3.1.2 Métriques de performance

On définit :

- *Délai de bout-en-bout* d'un paquet, comme étant le délai entre l'instant d'émission du paquet par la couche application émettrice jusqu'à la réception du paquet par la couche application réceptrice.

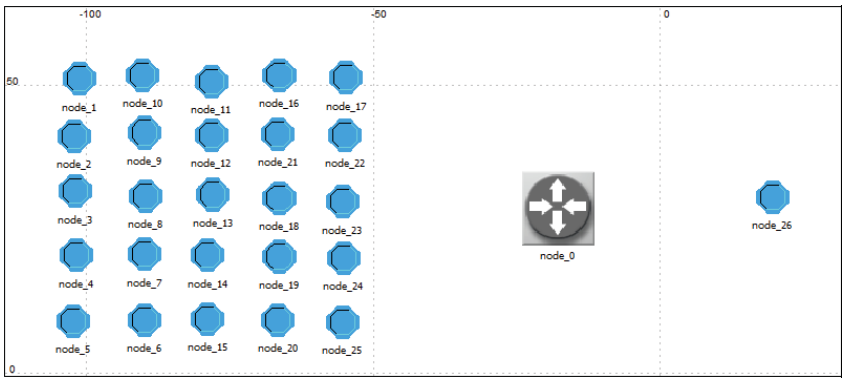


FIGURE 3.7 – Réseau à un seul saut.

- *Succès de transmission* d'un bit, comme étant une fonction qui vaut un si le bit est bien reçu par la couche application réceptrice, zéro sinon.
- *Charge* du réseau, comme étant le nombre de bits générés par les couches applications émettrices de tous les nœuds par unité de temps.

Afin d'analyser les performances de notre protocole, nous considérons les métriques suivantes.

- *Débit du système* : c'est le nombre de bits reçus par toutes les couches applications réceptrices du réseau par unité de temps.
- *Délai moyen de bout-en-bout :* c'est la moyenne des délais de bout-en-bout de tous les paquets reçus avec succès par les couches applications réceptrices.
- *Taux de succès de transmission :* c'est la somme des succès de transmissions de tous les bits générés par les couches applications émettrices divisé par le nombre total de ces bits.

Par la suite nous présentons les performances de CoSenS pour trois configurations de réseaux différentes. Dans chacune, nous étudions un ou plusieurs aspects de CoSenS.

3.3.2 Cas d'un réseau en étoile

3.3.2.1 Description du réseau et des scénarios

On considère le réseau en étoile illustré par la figure 3.7. Il est composé de 25 nœuds simples (disques bleus dans la figure) générant des données, un récepteur ("node_26") et un routeur ("node_0"). Les générateurs envoient des données vers le récepteur à travers le routeur. Le routage est statique. Les nœuds sont à un seul saut les uns des autres.

Nous comparons les performances de CoSenS avec IEEE 802.15.4 ainsi que les deux algorithmes d'estimation entre eux. Nous simulons les deux types de trafic, à savoir périodique et Poisson. Dans le cas du trafic périodique, nous simulons deux cas ; sans déphasage et avec déphasage. Dans le premier cas, tous les nœuds simples génèrent du trafic au même instant. Dans le deuxième cas, les instants de génération

sont différents pour chaque nœud. Cela est fait en tirant une valeur aléatoire entre $[t_{départ}, t_{départ} + 1]$. La durée simulée est égale à 15min.

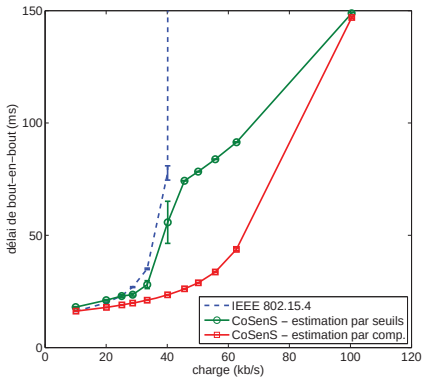
3.3.2.2 Résultats

Les figures 3.8, 3.9 et 3.10 montrent les résultats de simulation du réseau en étoile dans le cas du trafic Poissonien, périodique et périodique différé, respectivement. Globalement, CoSenS améliore significativement les performances de IEEE 802.15.4. Nous remarquons aussi que la performance de CoSenS s'améliore avec l'augmentation de la charge comparée à IEEE 802.15.4 et ceci pour tous les types de trafic. IEEE 802.15.4 sature pour une charge égale à 44 kb/s. Ses performances se dégradent significativement après. Quant à CoSenS, les performances commencent à se dégrader pour une charge supérieure à 62 kb/s. Cependant, la dégradation n'est pas significative ; les résultats de CoSenS restent largement supérieurs à ceux de IEEE 802.15.4. Par la suite, nous approfondissons l'analyse de ces résultats.

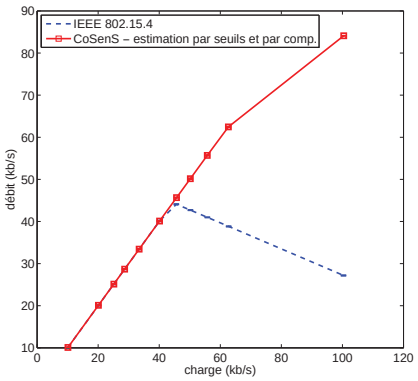
Les figures 3.8(a) et 3.10(a) illustrent le délai de bout-en-bout du trafic Poissonien et celui périodique différé. Nous remarquons que pour des charges faibles, les résultats de CoSenS et de IEEE 802.15.4 sont similaires avec un léger avantage pour le dernier. Après, lorsque la charge augmente, l'écart se creuse entre les deux protocoles, en faveur de CoSenS. En effet, en augmentant la charge, CoSenS collecte plus de paquets. Ainsi, le gain obtenu de la transmission en burst augmente. Les figures 3.8(d) et 3.10(d) illustrent les durées moyennes des périodes d'attentes. Nous constatons bien que cette durée augmente quand la charge augmente. Pour le trafic périodique, nous remarquons que le délai obtenu par CoSenS, Fig. 3.9(d), est toujours meilleur que celui de IEEE 802.15.4. Puisque le trafic est généré aux mêmes instants, le réseau est momentanément surchargé à ces instants-là. L'adaptation rapide de CoSenS à la charge, à cause du filtre non linéaire que nous avons utilisé, lui permet de s'adapter très rapidement à la situation et de collecter beaucoup de paquets comme le montre la figure des durées de la période d'attente, Fig. 3.9(d). Le gain de la transmission en burst est donc visible même pour des charges faibles du trafic.

La collecte de donnée, l'adaptation de la période d'attente et la transmission en burst améliore aussi significativement le débit du système (Fig. 3.8(b), Fig. 3.9(b) et Fig. 3.10(b)) et les taux de succès de transmission (Fig. 3.8(c), Fig. 3.9(c) et Fig. 3.10(c)). En effet, la combinaison de ces trois mécanismes permet d'éliminer le gaspillage de temps lié à l'utilisation de CSMA/CA pour chaque paquet. CoSenS est ainsi plus efficace en termes d'utilisation de bande passante.

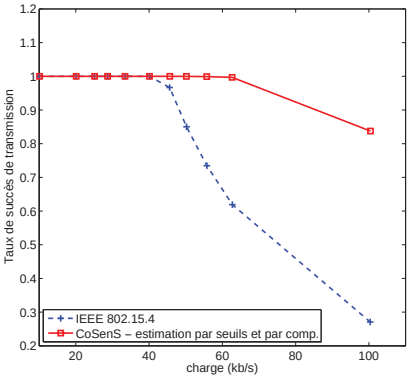
Pour les deux algorithmes d'estimation, nous remarquons que l'estimation par comparaison donne de meilleurs résultats grâce à une meilleure estimation de la durée d'attente. Ce qui réduit le temps d'attente inutile pendant la période d'attente et donc améliore l'efficacité du protocole.



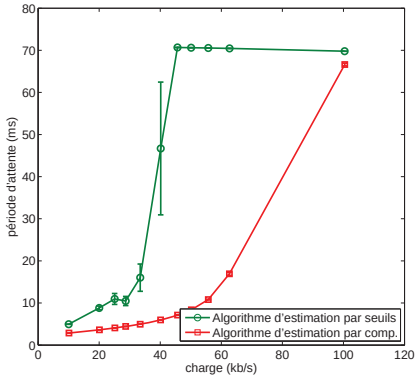
(a) Délai de bout-en-bout.



(b) Débit du système.

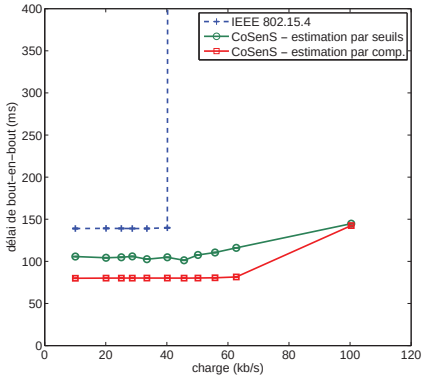


(c) Taux de succès de transmission.

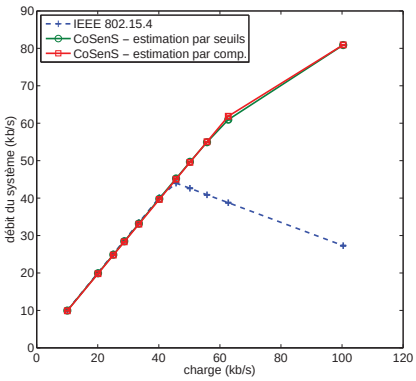


(d) Durée de la période d'attente dans CoSenS.

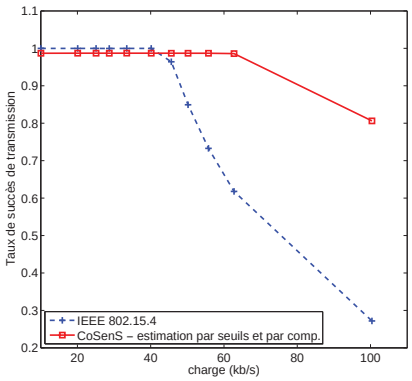
FIGURE 3.8 – Résultats de simulation du réseau en étoile dans le cas d'un trafic Poissonien. Intervalle de confiance à 95%.



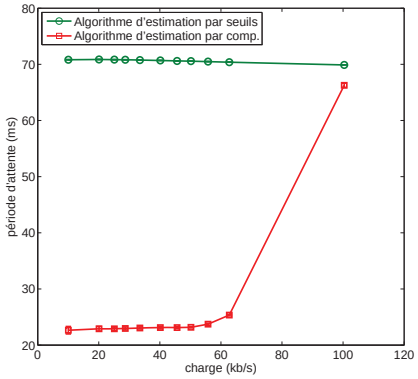
(a) Délai de bout-en-bout.



(b) Débit du système.

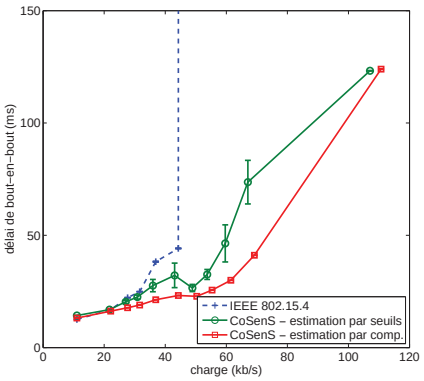


(c) Taux de succès de transmission.

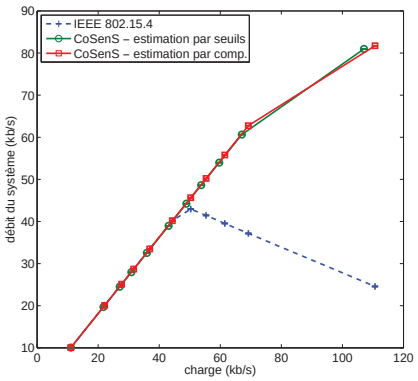


(d) Durée de la période d'attente dans CoSenS.

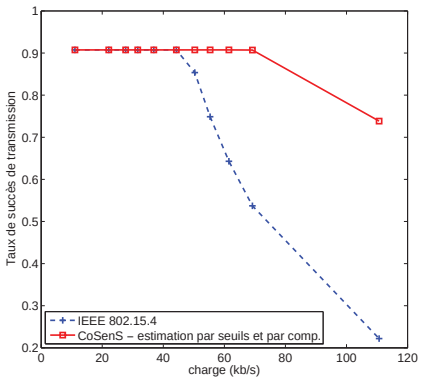
FIGURE 3.9 – Résultats de simulation du réseau en étoile dans le cas d'un trafic périodique. Intervalle de confiance à 95%.



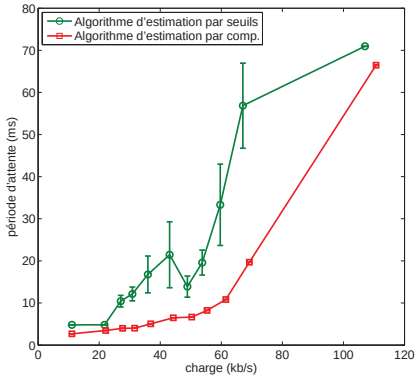
(a) Délai de bout-en-bout.



(b) Débit du système.



(c) Taux de succès de transmission.



(d) Durée de la période d'attente dans CoSenS.

FIGURE 3.10 – Résultats de simulation du réseau en étoile dans le cas d'un trafic périodique différé. Intervalle de confiance à 95%.

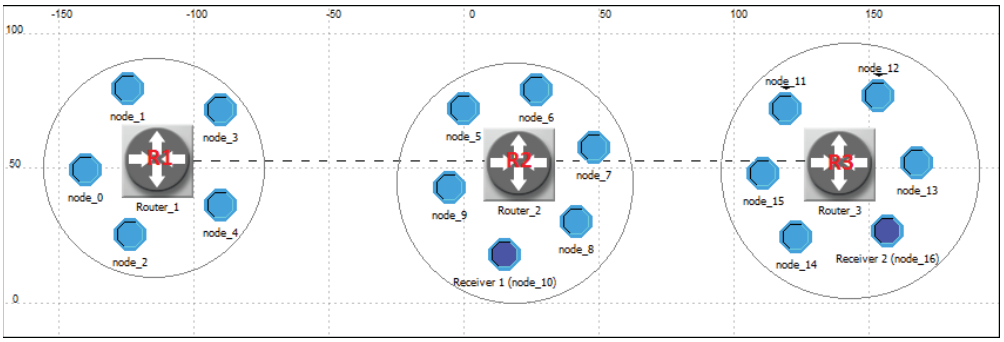


FIGURE 3.11 – Réseau en ligne.

3.3.3 Cas d'un réseau en ligne

3.3.3.1 Description du réseau et des scénarios

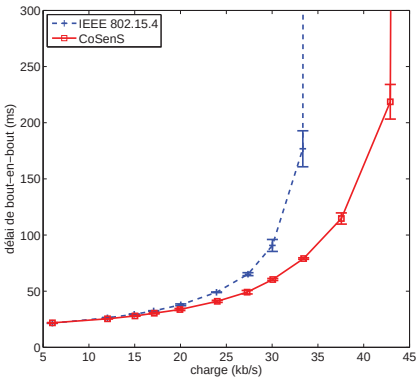
On considère un réseau composé de 15 nœuds simples, deux récepteurs et trois routeurs organisés en ligne comme le montre la figure 3.11. Chaque routeur écoute ses voisins immédiats dans la ligne. Les lignes en tirets donnent les liens sans fil entre ces routeurs. Les nœuds simples sont organisés en trois groupes, chacun attaché à un routeur. Un nœud simple ne peut entendre que le routeur auquel il est attaché et les nœuds simples membres du même groupe. Ce réseau simule un segment d'un réseau plus grand.

Deux scénarios sont simulés. Dans le premier, tous les nœuds simples envoient des données vers le récepteur 1 qui est attaché à R2. Ce scénario souffre du problème du terminal caché. Dans le deuxième scénario, tous les nœuds envoient des données vers le récepteur 2 qui est attaché à R3. Le groupe de nœuds simples attachés au routeur R1 envoie des données pendant toute la durée de la simulation. Les deux autres groupes envoient des données entre les instant 300 s et 600 s. Ce scénario permet de voir l'effet de l'adaptation de la période d'attente. Pour tous les scénarios, la durée simulée est égale à 900 s et la période d'attente est adaptée en utilisant l'algorithme d'estimation par comparaison.

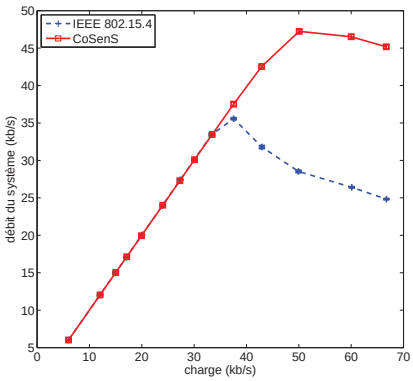
3.3.3.2 Résultats

La figure 3.12 montre les résultats du premier scénario. CoSenS obtient de meilleures performances comparé à IEEE 802.15.4. En effet, ce dernier sature pour une charge égale à 38 kb/s alors que CoSenS sature pour des charges supérieures à 47 kb/s. De plus, comme dans le réseau en étoile, les performances de CoSenS sont similaires à IEEE 802.15.4 pour les charges faibles et s'améliorent quand la charge du réseau augmente. Nous observons que les performances de CoSenS restent satisfaisantes en présence du problème du terminal caché. Cependant, un mécanisme similaire au RTS/CTS de IEEE 802.11 peut être utilisé pendant la période de transmission pour pallier ce problème. Notons que puisque les paquets à transmettre par un

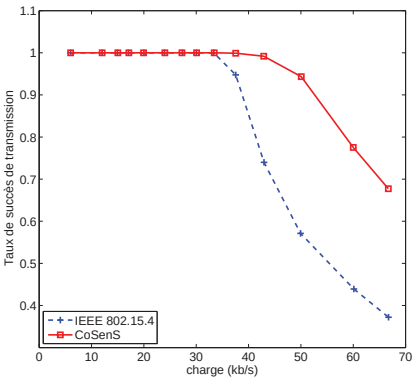
routeur n'ont pas forcément la même destination, il est nécessaire d'utiliser autant de messages RTS que de prochains sauts (ceci dit une exception peut être faite aux prochains sauts qui n'ont qu'un seul paquet à recevoir). L'introduction de ce mécanisme impose une modification supplémentaire à la période de transmission afin de préserver le mode de transmission en burst ; si le CTS n'est pas reçu après l'envoi d'un RTS à un prochain saut donné, les paquets à transmettre à ce prochain saut sont ordonnancés en dernier et une nouvelle tentative de transmission de ces paquets est faite à la fin de transmission des autres. Si le CTS n'est toujours pas reçu pendant la deuxième tentative, la transmission en burst s'arrête.



(a) Délai de bout-en-bout.



(b) Débit.



(c) Taux de succès de transmission.

FIGURE 3.12 – Résultats de simulation du premier scénario du réseau en ligne. Intervalle de confiance à 95%.

La figure 3.13 montre la charge en fonction du temps ainsi que les durées des périodes d'attente des trois routeurs en fonction du temps dans le cas du deuxième scénario. Nous remarquons bien que lorsque la charge augmente, entre les instants 300 s et 600 s, les périodes d'attentes s'adaptent très rapidement à cette augmentation. Notons aussi que c'est le routeur 3 qui enregistre l'augmentation la plus importante

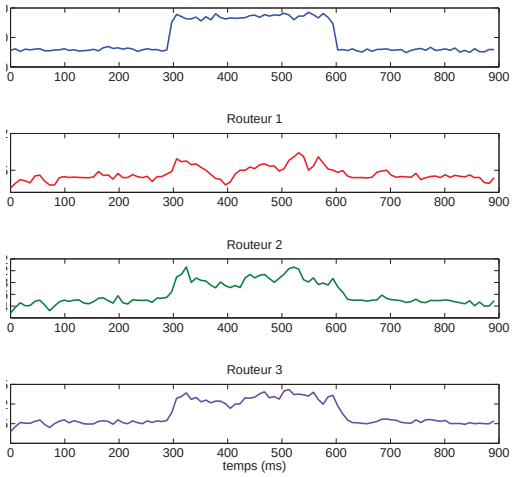


FIGURE 3.13 – Adaptation des périodes d'attente des routeurs dans le cas du deuxième scénario du réseau en ligne.

parce que la destination est un de ses fils et donc tout le trafic passe par lui pour l'atteindre. Lorsque la charge diminue, après 600 s, les durées de périodes d'attentes diminuent aussi.

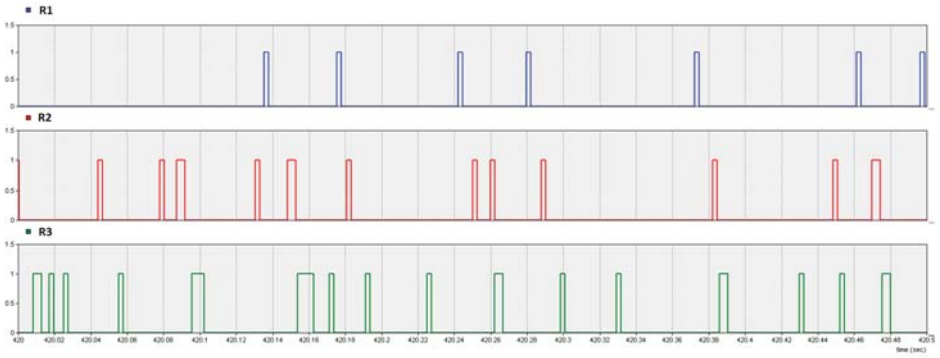


FIGURE 3.14 – Organisation des WP et des TP des trois routeurs pendant une demi seconde du temps simulé (420 s à 420.5 5). Zéro correspond à une WP et un à une TP.

La figure 3.14 montre comment les périodes d'attente et de transmission des trois routeurs sont organisées dans le cas du deuxième scénario. Les résultats sont tracés pour une demi seconde du temps simulé se trouvant dans la période de haute charge ($[300\,s, 600\,s]$). Nous observons que les périodes de transmission sont synchronisées ; celle du routeur R1 précède celle du routeur R2 qui à son tour précède celle de R3. De plus, nous remarquons que les périodes de transmission sont plus nombreuses pour le routeur R3. La destination étant un fils du routeur R3, celui-ci est donc

le plus chargé des trois. Les périodes de transmission ne se chevauchent pas au cours de cette demi-seconde, ce qui est le cas la plupart du temps pour ce scénario. Cependant, il est possible d'avoir des périodes de transmission qui se chevauchent. Un tel cas se produit lorsque la transmission en mode burst s'arrête pour un routeur et qu'un autre gagne l'accès au support de transmission après.

Plus généralement les périodes de transmission des routeurs sont synchronisés dans le cas où il y a un nombre limité de puits (idéalement un seul puits) et le routage s'effectue selon l'arbre. En effet, les durées des périodes d'attente augmentent lorsqu'on se rapproche du puits. Les périodes de transmission d'un routeur situé dans une profondeur k sont généralement incluses dans les périodes d'attentes du routeur père situé dans la profondeur $k-1$. De plus, le protocole CSMA/CA arbitre l'accès des routeurs voisins. Cela implique que leurs périodes de transmission ne se chevauchent pas sauf dans l'éventualité d'un échec de transmission.

3.3.4 Cas d'un réseau multi-saut en grille

3.3.4.1 Description du réseau et du scénario

On considère un réseau multi-saut composé de 25 routeurs et 125 nœuds déployés sur une surface de $1000\times1000\,m^2$. Les nœuds sont organisés en grille comme le montre la figure 3.15. Pour chaque carré de la grille on trouve un routeur, situé au centre, et cinq nœuds simples déployés aléatoirement dans ce carré. Le réseau utilise le R.A. de ZigBee, avec les paramètres $Cm = 12$, $Rm = 5$ et $Lm = 6$. Les lignes blanches dans la figure représentent un exemple d'association entre les routeurs. La durée de la simulation est égale à 16 min dont une qui est réservée à l'association. Ce réseau représente un système de surveillance de patients [Ko 2010] ou un système équivalent, où chaque nœud simple est attaché à un patient pour surveiller ses signes vitaux et chaque routeur est situé dans un secteur de l'hôpital. Tous les nœuds simples envoient des données vers le ROOT, situé au centre du réseau. Le trafic étant Poissonien.

Dans le reste du manuscrit, nous retenons ce scénario pour étudier les performances des différentes extensions du protocole CoSenS que nous proposons.

3.3.4.2 Résultats

La figure 3.16 présente les résultats de simulation du réseau en grille. Nous remarquons que les résultats sont similaires aux deux premiers scénarios. CoSenS et IEEE 802.15.4 atteignent la saturation pour des charges supérieures à 60 kb/s et 47 kb/s, respectivement. CoSenS améliore le délai de bout-en-bout et le taux de succès de transmission, deux paramètres importants pour ce type de réseau et application. Cela démontre que le comportement de CoSenS ne varie pas avec l'augmentation du nombre de nœuds.

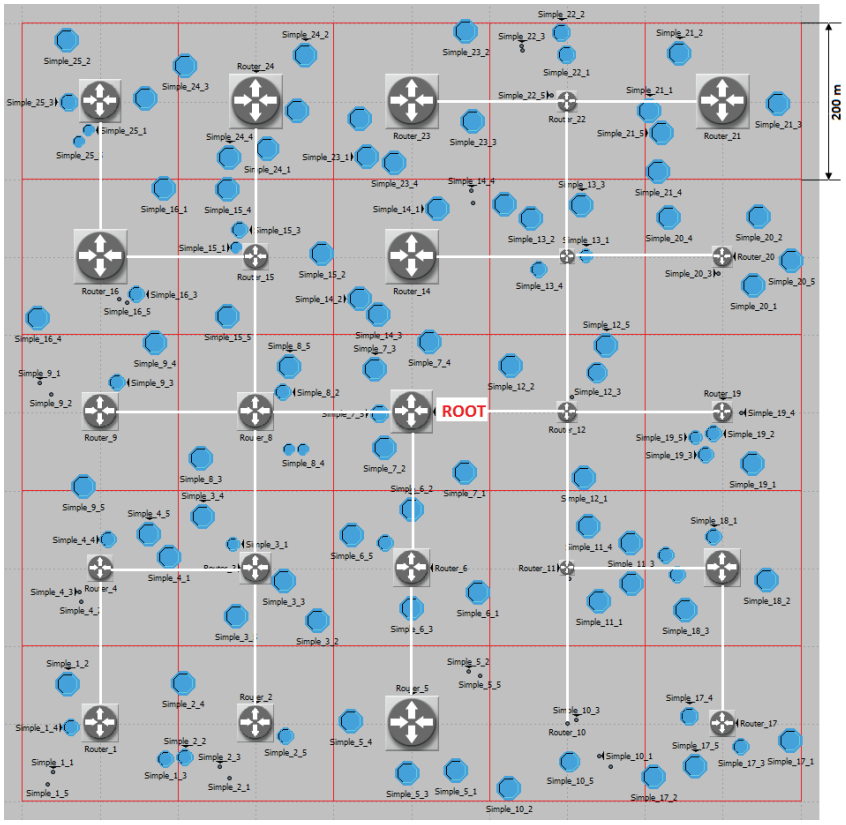
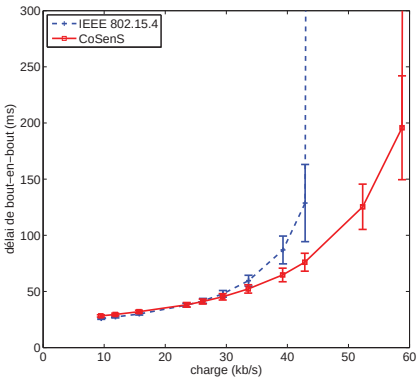
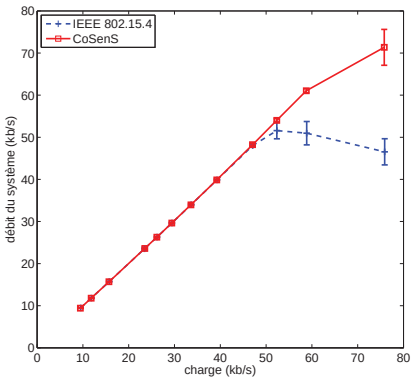


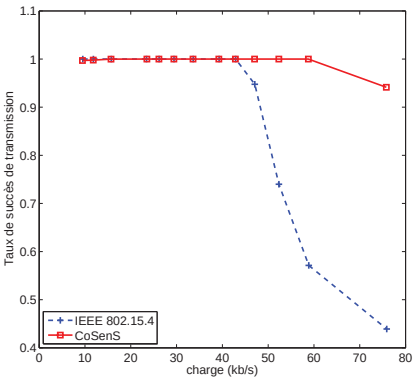
FIGURE 3.15 – Réseau multi-saut en grille. Intervalle de confiance à 95%.



(a) Délai de bout-en-bout.



(b) Débit.



(c) Taux de succès de transmission.

FIGURE 3.16 – Résultats de simulation du réseau grille.

3.4 Implémentation et expérimentation sur la plateforme Jennic

3.4.1 Implémentation

Nous avons implémenté CoSenS sur les capteurs sans fil Jennic JN5148 (Fig. 3.17). Le JN5148 est composé d'un processeur RISC 32-bits, 128 ko de ROM et 128 ko de RAM et opère dans la bande 2.4 GHz [Jennic Ltd. 2011d]. L'implémentation du standard IEEE 802.15.4 est fournie en tant qu'ensemble de librairies qui seront utilisées pendant la phase de développement. Une liste d'API est aussi fournie pour programmer et contrôler la couche MAC d'IEEE 802.15.4 et les différents périphériques du capteur. Cela implique que la couche MAC ne peut pas être modifiée. Ainsi, CoSenS est implémenté dans une couche intergicielle. Nous avons opté pour ce choix afin de comparer équitablement les performances de CoSenS avec CSMA/CA non slotté ainsi que de montrer qu'il peut être implémenté sans modification du

FIGURE 3.17 – Un capteur JN5148.

standard lui-même. Les détails de l'implémentation sont présentés dans l'annexe C. Les algorithmes d'estimation sont implémentés tel que présentés dans la section 3.2.5 car le processeur du JN5148 est capable de gérer les flottants. La couche MAC fournie par Jennic utilise toujours CSMA/CA pour transmettre des paquets de données. Ainsi, pour le désactiver afin de transmettre en mode burst (CSMA/CA est utilisé uniquement pour transmettre le premier paquet), les deux paramètres $macMinBE$ et $macMaxCSMABackoffs$ doivent être fixés à zéro. Cependant, nous les avons fixé à un car nous avons remarqué que le délai et le taux de perte de paquets sont plus importants lorsque ces paramètres sont fixés à zéro. Nous n'avons pas pu déterminer la cause avec exactitude en raison des spécificités de la plateforme Jennic.

3.4.2 Expérimentation

3.4.2.1 Réseau en étoile

Le scénario consiste en un réseau à un seul saut, composé de 10 nœuds simples générant des messages, un seul routeur qui agit comme coordinateur du PAN et un récepteur organisés comme le montre la figure 3.18.

TABLE 3.2 – Paramètres de l'expérimentation.

Variable		valeur
Taille du paquet	APL	96 bits
	PHY	216 bits
Nombre de paquets générés		1000
Type du trafic		periodique différé

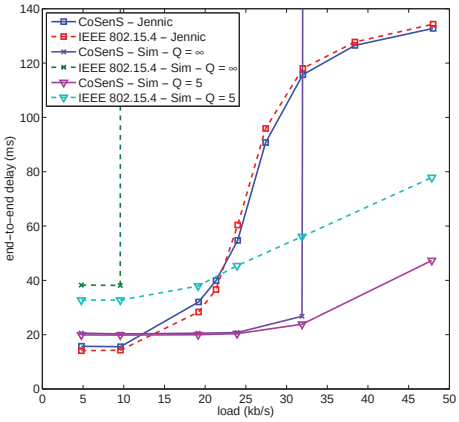
Les valeurs des paramètres de CoSenS et de IEEE 802.15.4 sont les mêmes que

FIGURE 3.18 – Scénario d'expérimentation.

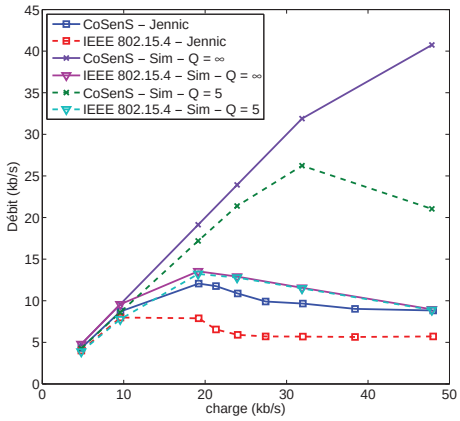
pour la simulation (Tab. 3.1). Le reste des paramètres est résumé dans Tab. 3.2. La taille du paquet dans l'expérimentation est égal à 96 bits au niveau de l'application, auxquels on ajoute 120 bit d'entêtes MAC et physique. Les 96 bits du paquet de l'application sont divisés en 8 bits pour le numéro de séquence, 8 bits pour le type de la trame, 16 bits pour l'adresse source, 16 bits pour l'adresse destination, 32 bits pour le temps de génération du paquet, 8 bits pour la mesure de la température et 8 bits pour la mesure de l'humidité. Pour l'analyse des performances, les deux derniers paramètres ont été remplacés par le numéro de paquet généré (sur 16 bits). Les nœuds simples génèrent un trafic périodique différé [3] (1 ms de décalage entre chaque nœud). Après l'association, le routeur diffuse un paquet qui signale le début de la transmission des messages de données. Chaque nœud simple envoie 1000 messages au récepteur à travers le routeur (comme dans le scénario à un seul saut dans la simulation). Nous varions les périodes de génération et nous mesurons les délais de bout-en-bout, le débit réseau et le taux de succès de transmissions. Tous les messages sont acquittés.

La figure 3.19 présente les résultats de l'expérimentation sur la plate-forme de capteurs Jennic ainsi que ceux de la simulation du même scénario avec des tailles de files d'attentes différentes. Pour le délai de bout-en-bout, Fig. 3.19(a), les résultats de CoSenS et de IEEE 802.15.4 sont similaires avec un léger avantage pour le dernier dans le cas de faibles charges, un avantage qui s'inverse au fur et à mesure que la charge augmente. Ces résultats peuvent être justifiés par trois facteurs. Premièrement, un peu de temps est gaspillé pendant la transmission des paquets en burst. Cela est dû à l'utilisation systématique de CSMA/CA pour la transmission

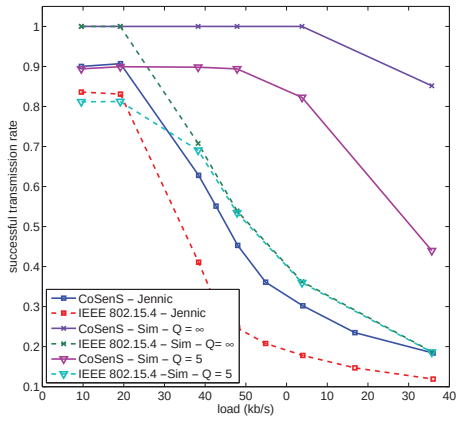
3. *i.e.* avec des instants de génération décalés



(a) Délai de bout-en-bout.



(b) Débit.



(c) Taux de succès de transmission.

FIGURE 3.19 – Résultats de l'expérimentation sur la plateforme de capteurs Jennic et de la simulation dans le cas d'un réseau à un seul saut.

et le fait qu'il n'est pas totalement désactivé. Deuxièmement, les files d'attente au niveau de la couche MAC ont une taille fixe[4], ce qui engendre un taux de perte de paquets plus important mais aussi des délai de bout-en-bout plus courts. Cette observation est confirmée par les résultats de la simulation du même scénario en utilisant une file infinie et une file de taille égale à cinq. Troisièmement, la couche MAC est implémentée comme une machine à états finis [Jennic Ltd. 2011a]. Ainsi, elle n'est pas capable de gérer les émissions et les réceptions au même temps, comme dans un système d'exploitation multi-tâche, contiki [Dunkels 2011] en l'occurrence. Pour le débit du système, Figure.3.19(b), CoSenS obtient de meilleurs performances qu'IEEE 802.15.4. Ce qui permet d'avoir des taux de succès de transmission plus importants (Figure.3.19(c)).

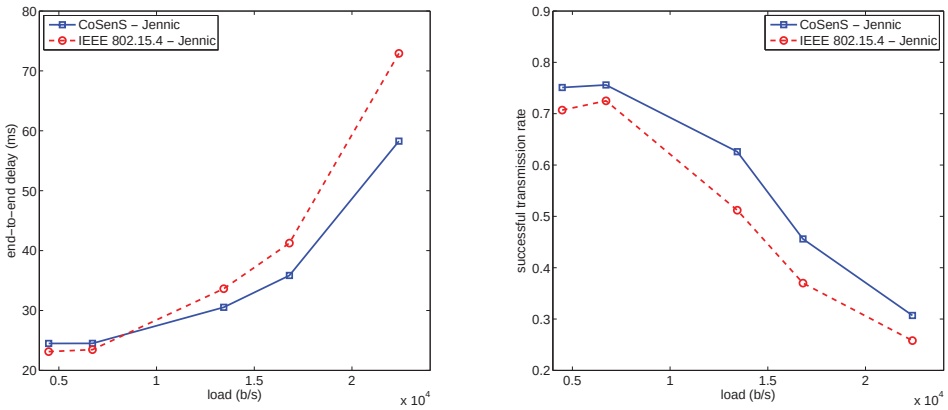
Dans les cas des capteurs Jennic, CoSenS améliore le débit du système sans dégrader les délais de bout-en-bout. Les résultats de CoSenS peuvent être améliorés s'il est implémenté directement dans la couche MAC. De cette façon, les files d'attentes de CoSenS et de transmission peuvent être gérées ensemble.

3.4.2.2 Réseau en ligne

Ce scénario est semblable à celui de la simulation (figure 3.11). Il est composé de trois routeurs, R1, R2 et R3 organisés en une ligne (R3 est attaché à R2 qui est lui-même attaché à R1 qui joue le rôle du coordinateur du PAN) et huit nœuds simples organisés comme suit. Trois nœuds simples sont attachés à R1 dont un qui joue le rôle du récepteur, trois nœuds simples sont attachés à R2 et deux nœuds simples sont attachés à R3. La distance maximale entre deux routeurs est égale à 25 m. Les nœuds simples, hormis le récepteur, génèrent un trafic périodique avec la même période. Chaque groupe de nœuds simples a un temps de génération décalé. Chaque nœud simple génère 1000 messages et les envoie au récepteur. On a varié les périodes de génération et on a mesuré le délai de bout-en-bout et le taux de succès de transmission. Les autres paramètres de l'expérimentation sont identiques au scénario du réseau en étoile.

La figure 3.20 présente les résultats de l'expérimentation du réseau en ligne sur la plateforme Jennic. Les résultats montrent que CoSenS obtient toujours de meilleurs taux de succès de transmission par rapport à IEEE 802.15.4. Pour les délais de bout-en-bout, CoSenS se comporte comme dans la simulation. Dans le cas de faibles charges, les performances de IEEE 802.15.4 sont légèrement meilleures que celles de CoSenS. La tendance s'inverse et l'écart se creuse en faveur de CoSenS lorsque la charge augmente.

4. L'information à propos de sa taille est indisponible sur Jennic.



(a) Délai de bout-en-bout. (b) Taux de succès de transmission.

FIGURE 3.20 – Résultats de l'expérimentation sur la plateforme de capteurs Jennic et de la simulation dans le cas d'un réseau en ligne.

3.5 Conclusion

Dans ce chapitre nous avons proposé un nouveau mécanisme auto-adaptatif, appelé CoSenS, qui permet d'améliorer les performances du protocole CSMA/CA non slotté en terme de délai de bout-en-bout, de débit réseau et de taux de succès de transmission. Le fonctionnement du protocole se base sur le principe de collecte puis l'envoi des données. La durée de la période d'attente pendant laquelle un routeur collecte les données dépend de l'historique de la charge reçue par ce routeur. Cela permet d'optimiser le temps d'attente et même d'améliorer le délai de bout-en-bout. En effet, l'optimisation de la durée d'attente et la transmission en burst des données reçues permettent à CoSenS de réaliser des gains de performance en termes de débit et de délai de bout-en-bout.

En collectant les données, le routeur a la possibilité d'ordonnancer les paquets en utilisant n'importe quel algorithme d'ordonnancement. En effet, l'un des problèmes empêchant l'utilisation de politiques d'ordonnancement pour la différenciation de service dans les réseaux sans fil est le fait que le routeur n'a pas une idée précise sur les flux à ordonnancer. Cela est dû au média sans fil qui est partagé par l'ensemble des nœuds y compris le routeur lui-même. A chaque instant un et un seul nœud peut émettre et tout le reste peut recevoir. La collecte de donnée permet au routeur de déterminer la priorité de chaque paquet reçu et ainsi de les classer plus efficacement. Les autres avantages de la collecte de données incluent la possibilité de les agréger, ce qui améliore le débit, et d'effectuer le contrôle de congestion plus efficacement puisqu'il a une connaissance totale sur les paquets reçus. En combinant ces avantages avec l'amélioration des performances de CSMA/CA, CoSenS constitue le mécanisme de base pour le support de la QdS.

Chapitre 4

P-CoSenS : CoSenS avec priorité pour la différentiation de service

Sommaire

Ce chapitre présente P-CoSenS, une extension de CoSenS qui traite le problème de la différentiation de service. Notre solution se base sur l'utilisation de politiques d'ordonnancement, qui sont implémentées au dessus de CoSenS.

Le chapitre est organisé comme suit. Dans la première section, nous motivons les choix effectués. Dans la deuxième section, nous détaillons les trois politiques d'ordonnancement que nous avons retenues. La section 4.3 présente l'évaluation de performances de notre solution. La dernière section conclut le chapitre.

4.1 Motivations

La différentiation de service est l'un des mécanismes de base de la QdS. Elle prend tout son sens lorsque le réseau est amené à gérer plusieurs classes de trafic, qui ont chacune leurs exigences. Tel est le cas pour les réseaux de capteurs sans fil. En effet, les flux de données peuvent être plus ou moins urgents à transmettre. Il est clair qu'implémenter un mécanisme de différentiation de service est une nécessité afin de gérer ces différentes classes de flux. La plupart des solutions retenues dans les réseaux de capteurs sans fil se basant sur CSMA/CA, à l'instar de celle que nous avons proposée dans [Koubaa 2006], utilisent des paramètres différents de CSMA/CA en fonction de la nature du trafic. Cependant, comme démontré dans le chapitre

précédent, les performances de CSMA/CA se dégradent au fur et à mesure que la charge augmente. Les mécanismes de QdS doivent fonctionner correctement avec peu de dépendance vis-à-vis de la charge du réseau.

L'utilisation des politiques d'ordonnancement comme dans les réseaux filaires n'est pas efficace à cause du médium de transmission sans fil qui, par nature, diffuse physiquement l'information. Cela implique que le nœud sur lequel un tel mécanisme d'ordonnancement est implémenté, n'a pas une idée précise sur les paquets de données à ordonnancer, puisqu'ils sont transmis un par un. On considère ici que tous les nœuds utilisent une seule fréquence pour communiquer. En collectant les données avant de les transmettre, CoSenS résout ce problème, ce qui améliore l'efficacité des politiques d'ordonnancement. Ainsi, nous avons choisi d'utiliser des politiques d'ordonnancement pour la différentiation de service au niveau des routeurs. Toutefois, utiliser des paramètres différents de CSMA/CA selon la nature du trafic conjointement avec les politiques d'ordonnancement est possible. Ce mécanisme peut être implémenté dans les nœuds simples. La seule contrainte est de préserver la priorité des routeurs sur les nœuds simples. Dans ce chapitre, on se limite à l'étude des politiques d'ordonnancement.

4.2 Politiques d'ordonnancement de files d'attentes

4.2.1 Priorité fixe

Dans la politique d'ordonnancement à priorité fixe (PF) [Liu 1973], on considère deux types de trafic ; périodique et événementiel. Ainsi, on a deux priorités. On suppose que le trafic événementiel est plus prioritaire. Généralement, un événement correspond à une mesure du phénomène physique différente de sa valeur normale ou souhaitée. De ce fait, les événements doivent être transmis le plus rapidement possible et d'une manière aussi fiable que possible. Les données périodiques correspondent à des mises à jour régulières des mesures du phénomène physique et qui signalent généralement des fluctuations mineures des valeurs de ces mesures.

Les données événementielles sont donc transmises en premier. La transmission des paquets classés comme périodiques se fait après la transmission de tous les paquets classés comme événements. Pour chaque classe de trafic, les paquets sont servis selon leur ordre d'arrivée.

4.2.2 Earliest Deadline First

Quelques applications des réseaux de capteurs sans fil imposent une échéance sur le délai de bout-en-bout. Dans ce cas, la politique "Earliest Deadline First" (EDF) est plus adaptée à ce type d'applications.

On définit t_p^{rem} comme le temps restant avant que l'échéance du paquet p arrive à son terme. Elle est incluse dans l'entête du paquet. L'application du nœud source,

initialize cette variable avec l'échéance imposée au paquet. Après, t_p^{rem} est mise à jour à chaque saut pour prendre en compte les délais de séjour dans la file d'attente, de la contention pour accéder au medium et être servi. On définit, $prio_p^R$ comme étant la priorité du paquet p calculée par le routeur R. Elle est donnée par l'équation 4.1

$$prio_p^R = \frac{1}{t_p^{rem}} \tag{4.1}$$

Dès la réception du paquet par un routeur, ce dernier calcule sa priorité et l'insère dans la file d'attente selon cette priorité ; la plus grande valeur est placée en premier. Ainsi, la durée de séjour du paquet dans ce routeur, somme des délais de séjours dans la file d'attente et du temps de service, n'est pas immédiatement prise en compte. t_p^{rem} est mise à jour par le routeur lorsqu'il gagne l'accès au support et est sur le point de transmettre le paquet. Le fonctionnement est comme suit. D'abord, il marque le temps de réception du paquet par la couche MAC et le temps de début de service du même paquet. La différence entre les deux temps correspond à la durée de séjour dans la file d'attente, notée t_p^{qdel}. Ensuite, le routeur calcule le délai de contention (causé par CSMA/CA par exemple), noté t_p^{cont}, ainsi que le délai de transmission, noté t_p^{tr}. Dans le cas d'une retransmission, à cause de la perte du paquet ou de son acquittement, le routeur doit mettre à jour t_p^{cont} et t_p^{tr} pour prendre en compte ces délais additionnels. Plus particulièrement, la durée de transmission de l'acquittement et le temps d'attente avant la réception de l'acquittement doivent être ajoutés à la durée de contention, t_p^{cont}. Les délais de propagation sont ignorés car leurs valeurs sont négligeables par rapport aux délais de séjour dans la file d'attente et de transmission. Finalement, t_p^{rem} est mise à jour comme le montre l'équation 4.2.

$$t_p^{rem} = t_p^{rem} - t_p^{qdel} - t_p^{cont} - t_p^{tr} \tag{4.2}$$

Notons que les paquets qui n'ont pas fourni d'échéances ou l'ont raté seront ordonnancés à la fin de la file d'attente selon leur ordre d'arrivée ce qui est différent de la version classique de EDF où les paquets qui ont raté leurs échéances peuvent encore être ordonnancés près de la tête de la file d'attente. On remarque aussi que t_p^{rem} est calculée localement au niveau de chaque nœud et ne nécessite donc aucune synchronisation globale des horloges.

4.2.3 D2PQ : Distance Deadline Priority Queueing

La politique D2PQ permet un ordonnancement selon la distance et l'échéance du paquet. Elle prend en compte le temps restant avant que l'échéance du paquet arrive à son terme, t_p^{rem}, ainsi que la distance restante à parcourir, d_p^{rem}, qui se mesure en nombre de sauts. On définit $prio_p^R$ comme étant la priorité du paquet p calculée par le routeur R. Elle est donnée par l'équation 4.3, où $\frac{1}{\mu_p}$ est le temps de service du paquet p. t_p^{rem} est mise à jour de la même façon que EDF. d_p^{rem} est intégrée dans l'entête du paquet et mise à jour à chaque routeur.

$$prio_p^R = \frac{d_p^{rem} \cdot \frac{1}{\mu_p}}{t_p^{rem}} \tag{4.3}$$

Dès la réception du paquet par un routeur, ce dernier vérifie si l'inégalité 4.4 est respectée. Si ce n'est pas le cas, le paquet est ordonnancé à la fin de la file car il est impossible qu'il atteigne la destination avant que son échéance arrive à son terme. Si l'inégalité est respectée, le routeur calcule la priorité du paquet et l'insère dans la file d'attente selon cette priorité ; la plus grande valeur est placée en premier. D2PQ prend en compte aussi le temps de service d'un paquet. Cela permet un ordonnancement plus fin des paquets reçus.

$$Condition\ d'ordonnancement :\ t_p^{rem} > d_p^{rem} \cdot \frac{1}{\mu_p} \tag{4.4}$$

Exemple

Considérons deux paquets p_1 et p_2 qui ont la même taille (donc même temps de service), des temps restants avant l'échéance t_1 et t_2, des distances restantes à parcourir d_1^{rem} et d_2^{rem} et qui arrivent au même routeur R.
- Si $d_1^{rem} = d_2^{rem}$ alors D2PQ est équivalente à la politique EDF.
- Si $t_1 \approx t_2$ et $d_1^{rem} > d_2^{rem}$ alors p_1 est plus prioritaire que p_2 à condition que t_1 suffise pour atteindre la destination sans que l'échéance ne soit dépassée.
- Si $t_1 > t_2$ et $d_1^{rem} > d_2^{rem}$ alors n'importe quel ordonnancement fait l'affaire à condition que le temps restant pour les deux paquets suffise pour atteindre la destination sans que l'échéance ne soit dépassée.
- Si $t_1 < t_2$ et $d_1^{rem} > d_2^{rem}$ alors p_1 est plus prioritaire que p_2 à condition que t_1 suffise pour atteindre la destination sans que l'échéance ne soit dépassée.

4.3 Analyse de performances

4.3.1 Scénarios et métriques de performance

On considère le réseau simulant un système de surveillance de patients présenté dans 3.3.4. La figure 4.1 reprend le réseau simulé. Deux séries de simulations ont été réalisées pour ce réseau. Dans la première, trois scénarios ont été simulés. Dans chacun d'entre eux, un groupe constitué de 10%, 25% et 50% de l'ensemble des nœuds simples est sélectionné aléatoirement et génère un trafic de type Poisson avec un temps d'inter-arrivé égal à une seconde. Les nœuds restant génèrent du trafic périodique. Le trafic de type Poisson est considéré comme trafic événementiel. Il est donc plus prioritaire. Pour chaque scénario, cinq simulations ont été réalisées et dans chacune d'entre elle un ensemble différent de nœuds est choisi aléatoirement. C'est la moyenne de ces résultats qui est présentée. Dans tous les scénarios, on varie l'intensité du trafic périodique et on compare les performances de CoSenS et IEEE

802.15.4 implémentant l'ordonnancement PAPS [1], PF et EDF. Dans cet ensemble de simulation, tous les nœuds simples envoient des données vers le ROOT, situé au centre du réseau. Dans la deuxième série de simulation, on compare les performances de CoSenS et de IEEE 802.15.4 utilisant EDF et D2PQ dans le scénario suivant. Pour chaque événement généré par un nœud simple, le coordinateur du réseau (ROOT) lui envoie un message de réponse, considéré aussi comme un événement avec une échéance égale à celle de l'événement initial. Ces messages représentent les actions que le responsable de la surveillance effectue en réponse à cet événement. Notons que CoSenS utilise l'algorithme d'estimation par comparaison.

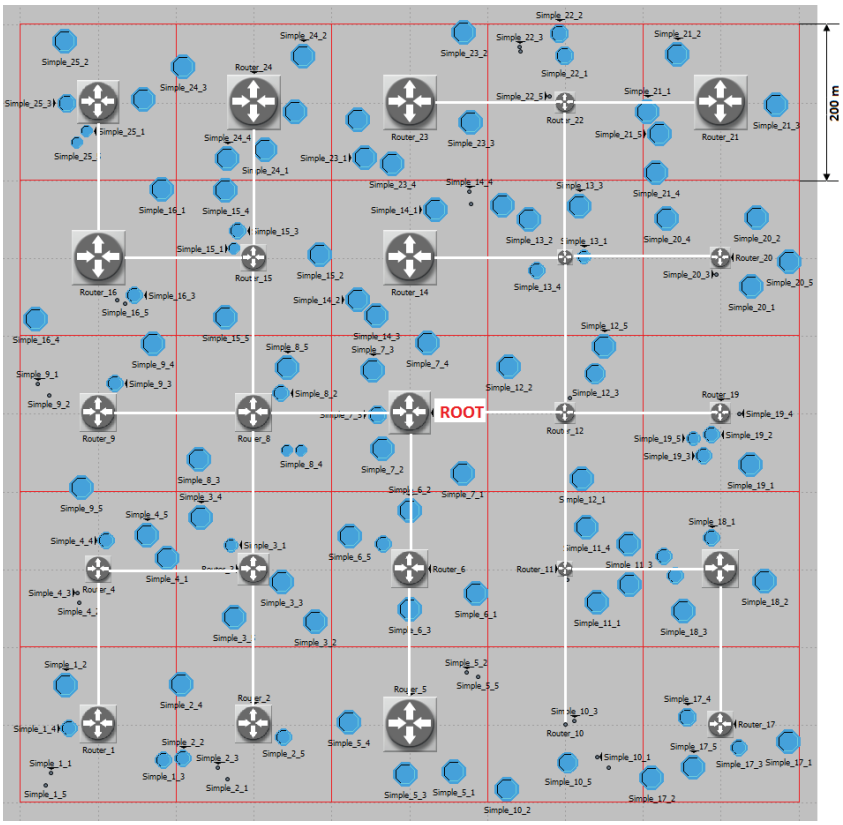


FIGURE 4.1 – Réseau multi-saut en grille.

Les métriques utilisées pour l'analyse de performance sont le délai de bout-en-bout, le taux de succès de transmission (TST) et le taux d'échéances respectées (TER). Les deux premières métriques ont été définies dans 3.3.1.2. On définit l'*échéance respectée* d'un paquet comme étant une fonction qui vaut un si le délai de bout-en-bout du paquet est inférieur à l'échéance fixée par l'application et zéro sinon. Le *Taux d'échéances respectées* est donc la somme des échéances respectées de tous les paquets reçus par les couches applications réceptrices divisé par le nombre total de ces paquets. Les résultats sont tracés en fonction de la charge totale générée par les

1. Premier Arrivé Premier Servi

couches application de tous les nœuds simples (somme de la charge événementielle et périodique).

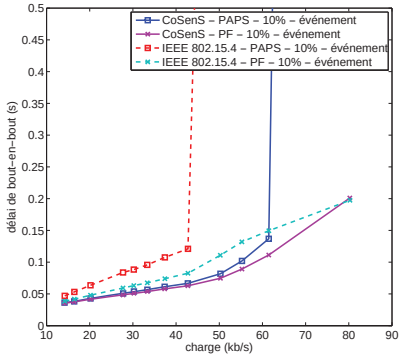
4.3.2 Résultats du premier ensemble de simulation

Les figures 4.2(a), 4.2(b) et 4.2(c) montrent les délais de bout-en-bout du trafic événementiel pour les trois scénarios. On observe que les performances du trafic événementiel sont meilleures lorsqu'on utilise la politique d'ordonnancement PF par rapport à celle PAPS et ce pour les deux protocoles. Plus spécifiquement, IEEE 802.15.4 avec ordonnancement PF améliore nettement les délais de bout-en-bout du trafic événementiel. En effet, lorsque la charge augmente, les routeurs utilisant IEEE 802.15.4 deviennent des goulots d'étranglement du réseau car ils n'arrivent plus à transmettre facilement tous les paquets qui leurs sont transmis. Ainsi, la taille de la file d'attente devient importante, ce qui accroît considérablement les délais de séjour dans les files d'attentes. Puisque le nombre de paquets événementiels est relativement petit par rapport à celui des paquets périodiques et ils sont insérés en tête de la file d'attente, leur délai de bout-en-bout est beaucoup plus petit. Cependant, lorsque le nombre de nœuds générant des événements augmente (scénarios 25% et 50%), la performance de IEEE 802.15.4 diminue car le nombre de paquets événementiels augmente et donc ils commencent à souffrir du problème du délai de séjour dans la file d'attente des routeurs.

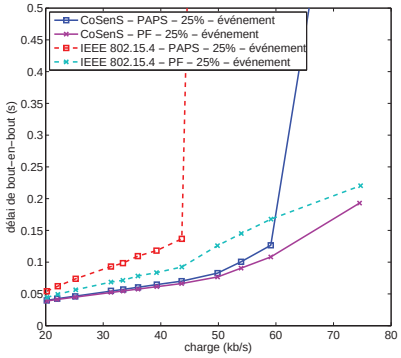
CoSenS utilisant PF obtient dans la plupart des cas de meilleurs résultats que IEEE 802.15.4. De plus, il améliore les délais du trafic événementiel par rapport à la version PAPS. Le seul cas où IEEE 802.15.4 utilisant PF obtient un meilleur résultat par rapport à CoSenS est lorsque celui ci est saturé. Ce phénomène est observé dans le cas du premier scénario pour une charge totale supérieure à 80 kb/s. On observe que contrairement à IEEE 802.15.4, les délais du trafic événementiel s'améliorent lorsque sa charge augmente (scénarios 25% et 50%). Ce résultat est dû à deux facteurs. Premièrement, les performances de CoSenS sont excellentes en dessous de sa charge de saturation. Deuxièmement, la charge totale du réseau descend en dessous de la charge de saturation de CoSenS lorsque la charge du trafic événementiel augmente. Cela est dû à deux facteurs. Le premier est que le trafic événementiel est généré avec un temps d'inter-arrivée constant. Le deuxième est que le nombre de nœuds générant un traffic périodique, qu'on varie, diminue (le 10%, 25% et 50% correspond à la fraction des nœuds qui génère un trafic événementiel).

Les figures 4.2(d), 4.2(e) and 4.2(f) présentent les délais de bout-en-bout du trafic périodique pour les trois scénarios. On observe que l'utilisation de la politique d'ordonnancement PF ne dégrade pas significativement la performance du trafic périodique. On remarque aussi que les résultats de CoSenS sont meilleurs que ceux de IEEE 802.15.4. Ainsi, CoSenS améliore globalement les délais de bout-en-bout du trafic événementiel même lorsque sa charge augmente tout en dégradant faiblement ceux du trafic périodique.

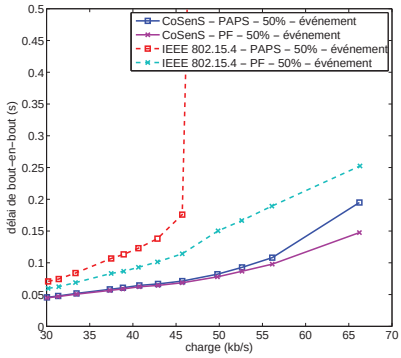
Les résultats de TST (Taux de Succès de Transmission) pour le trafic événementiel



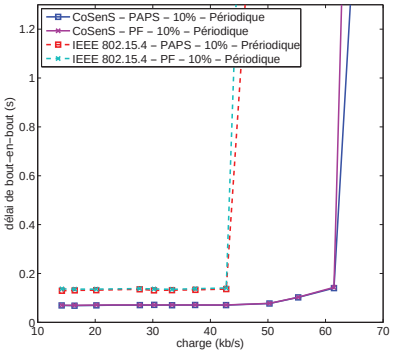
(a) Délai du trafic événementiel - scénario 10%.



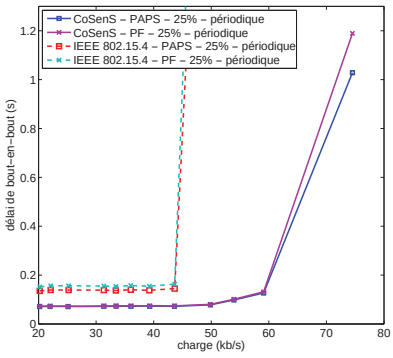
(b) Délai du trafic événementiel - scénario 25%.



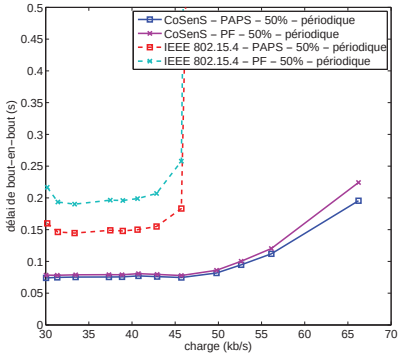
(c) Délai du trafic événementiel - scénario 50%.



(d) Délai du trafic périodique - scénario 10%.

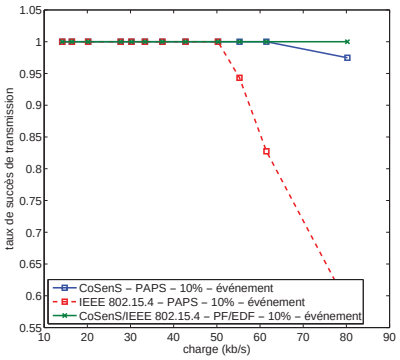


(e) Délai du trafic périodique - scénario 25%.

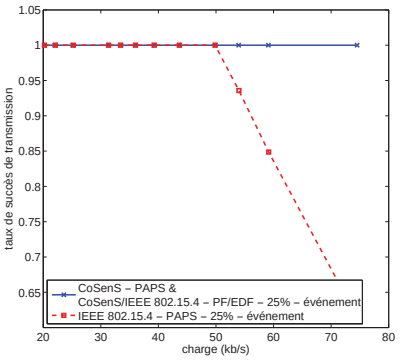


(f) Délai du trafic périodique - scénario 50%.

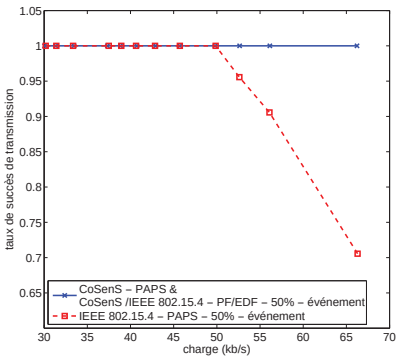
FIGURE 4.2 – Comparaison des délais de transmission de bout-en-bout du trafic événementiel et périodique entre CoSenS and IEEE 802.15.4 utilisant l'ordonnancement PAPS et PF.



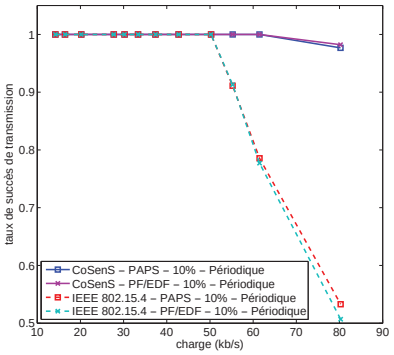
(a) TST du trafic événementiel - scénario 10%.



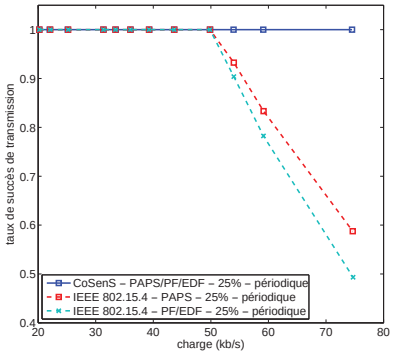
(b) TST du trafic événementiel - scénario 25%.



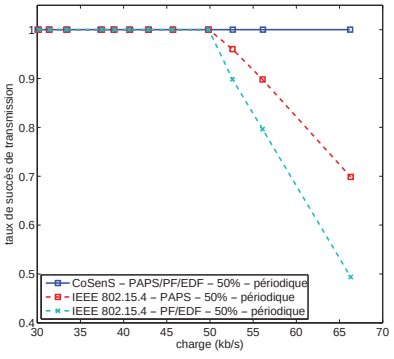
(c) TST du trafic événementiel - scénario 50%.



(d) TST du trafic périodique - scénario 10%.



(e) TST du trafic périodique - scénario 25%.

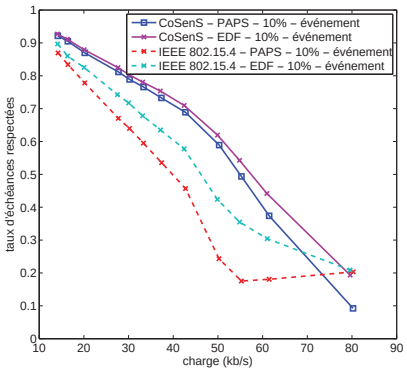


(f) TST du trafic périodique - scénario 50%.

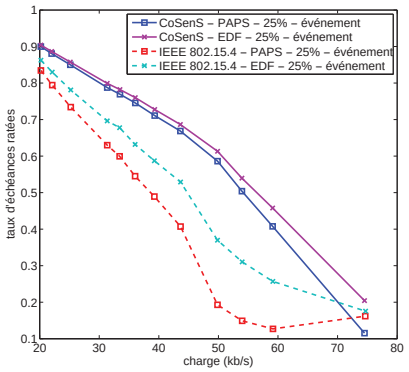
FIGURE 4.3 – Comparaison des taux de succès de transmission (TST) entre CoSenS IEEE 802.15.4 utilisant l'ordonnancement PAPS, FP et EDF. Les points qui sont supérieurs à 0.99 on été arrondi à 1.

sont illustrés dans les figures 4.3(a), 4.3(b) and 4.3(c). En utilisant PF et EDF, CoSenS et IEEE 802.15.4 améliorent grandement les résultats de TST par rapport à la politique d'ordonnancement PAPS. En effet ils obtiennent des valeurs proches de 100% de TST dans tous les scénarios de simulation. Cependant, pour IEEE 802.15.4 ce résultat est réalisé au prix d'une dégradation supplémentaire de la performance du trafic périodique. En effet, on observe dans les figures 4.3(d), 4.3(e) and 4.3(f) qu'en augmentant le trafic événementiel, le TST du trafic périodique obtenu par IEEE 802.15.4 diminue et ce après la saturation. Avant la saturation, les acquittements permettent d'obtenir des TST proches de 100%. Quand à CoSenS, il sature pour des charges supérieures à 80 kb/s. Ainsi, il est capable de réaliser de bonnes performances pour les deux types de trafic.

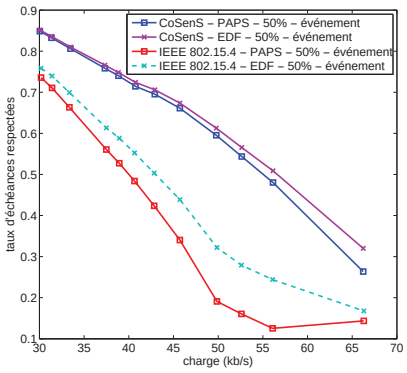
CoSenS utilisant PF améliore globalement les performances du trafic événementiel même lorsque sa charge augmente tout en dégradant faiblement les performances du trafic périodique.



(a) Scénario 10%.



(b) Scénario 25%.



(c) Scénario 50%.

FIGURE 4.4 – Taux d'échéances respectées (TER) de CoSenS de IEEE 802.15.4 utilisant l'ordonnancement EDF.

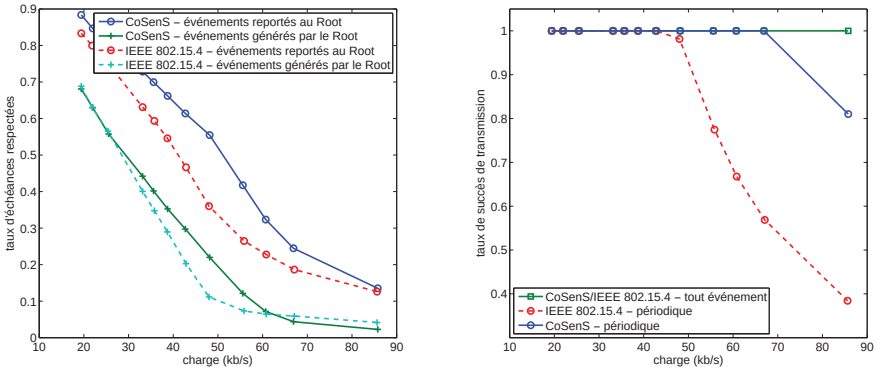
Afin d'étudier le TER (Taux des Échéances Respectées), on a fixé l'échéance sur les paquets événementiels à 0.08 s. Ce temps correspond au délai moyen de bout-en-bout obtenu par le trafic événementiel en utilisant IEEE 802.15.4 dans le cas où 50% des nœuds génèrent ce trafic (scénario 50%). La figure 4.4 montre les TER obtenus par CoSenS et par IEEE 802.15.4. On remarque que la politique d'ordonnancement EDF améliore le TER du trafic événementiel pour les deux protocoles. Pour IEEE 802.15.4 avec PAPS, on remarque que pour des charges importantes, le TER augmente. Ce résultat est dû à la baisse significative du nombre de paquets reçus par la couche application du ROOT (baisse du TST). Ainsi, les paquets qui sont transmis avec succès au ROOT, auront plus de chance de respecter leurs échéances. On rappelle que le TER est calculé par rapport au nombre de paquets correctement reçus. On observe aussi que la performance de CoSenS utilisant l'ordonnancement PAPS est meilleure que celle de IEEE 802.15.4 utilisant l'ordonnancement PAPS sauf lorsque CoSenS atteint le point de saturation. Ce résultat est causé par le faible TST obtenu par IEEE 802.15.4 par rapport à CoSenS et ce pour les deux types de trafic. Cependant, le fait que CoSenS donne de bons TST pour le trafic périodique, même après le point de saturation, peut affecter le TER du trafic événementiel, ce qui n'est pas souhaitable. Ainsi, un mécanisme de contrôle de congestion doit être implémenté afin d'éviter une telle situation.

CoSenS utilisant l'ordonnancement EDF améliore les résultats de CoSenS utilisant PAPS et IEEE 802.15.4 utilisant EDF. En effet la différence des TST entre les deux protocoles augmente avec l'augmentation de la charge totale. L'amélioration du délai de bout-en-bout avec l'augmentation de la charge en est la cause. On observe aussi que lorsque la charge événementielle augmente (trois scénarios), la performance de CoSenS s'améliore.

4.3.3 Résultats du deuxième ensemble de simulation

Dans cette série de simulation, on fixe l'échéance sur les paquets événementiels et les paquets de réponse à 0.08 s. On considère qu'un groupe formé par 10% de nœuds simples est aléatoirement choisi et génère un événement en suivant la loi de Poisson. Les TER des trafics événementiel et de réponse obtenus en utilisant l'ordonnancement EDF sont présentés dans la figure 4.5. On observe que la tendance est similaire aux résultats de TER pour le premier scénario (10%) de la première série de simulation. Sous son point de saturation, CoSenS utilisant l'ordonnancement EDF est meilleur qu'IEEE 802.15.4 utilisant l'ordonnancement EDF et ce pour les deux trafics événementiels ; celui généré par les nœuds et les réponses générées par le ROOT. Après la saturation, les performances de CoSenS se dégradent, à cause des TST élevés du trafic périodique (4.5(b)) qui affecte le TER du trafic événementiel.

La figure 4.6 présente les résultats obtenus par CoSenS en utilisant EDF et D2PQ. On observe que D2PQ améliore légèrement les taux d'échéances respectées et ce pour le trafic événementiel et de réponse. L'utilisation de la distance permet donc de raffiner les décisions d'ordonnancement sans pour autant augmenter la complexité



(a) TER des événements générés par/vers le ROOT.

(b) TST des données événementielles et périodiques.

FIGURE 4.5 – Comparaison des taux d'échéances respectées (TER) et taux de succès de transmission (TST) entre CoSenS et IEEE 802.15.4 utilisant l'ordonnancement EDF pour le trafic événementiel et de réponse.

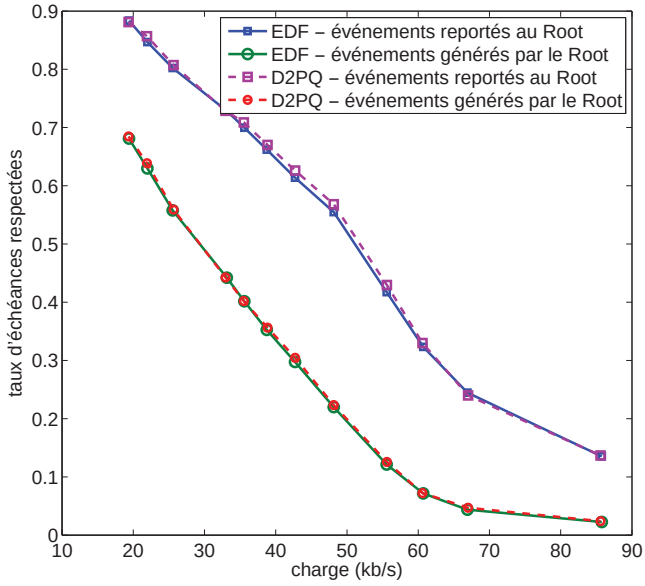


FIGURE 4.6 – TER de P-CoSenS utilisant les politiques d'ordonnancement EDF et D2PQ.

de cette opération.

La comparaison entre les TERs des deux trafics événementiels montre la limite de EDF et de D2PQ. En effet, le TER du trafic de réponse est inférieur à celui généré par les nœuds simples. Cela est dû au temps restant avant que l'échéance du paquet

arrive à son terme. Il diminue de plus en plus rapidement lorsque la charge augmente et les paquets convergent vers le ROOT. Ainsi, comparé au temps restant avant que l'échéance d'un paquet de réponse arrive à son terme, un paquet événementiel va être placé en tête de la file d'attente. De plus, les résultats de D2PQ montrent que le poids de la distance restante à parcourir ne permet pas de pallier ce problème car lorsque le temps restant avant l'échéance tend vers zéro (ce qui est le cas pour les événements qui convergent vers la destination), son inverse tend vers l'infini. Ainsi, le poids de la distance restante (six au maximum dans notre cas) ne permet pas aux messages de réponses d'être placés en tête de la file. Cela entraîne un délai de transmission plus long et donc des TER plus petits. Une solution possible serait d'intégrer à l'équation de calcul de la priorité (EDF ou D2PQ) un paramètre supplémentaire correspondant au type de message. Le calcul de la priorité pour la politique D2PQ peut être formulé selon l'équation 4.5 où β_p est un paramètre qui dépend de l'importance du paquet à transmettre.

$$prio_p^R = \beta_p + \frac{d_p^{rem} \cdot \frac{1}{\mu_p}}{t_p^{rem}} \tag{4.5}$$

4.4 Conclusion

Dans ce chapitre nous avons évalué les performances de deux classes de trafic ordonnancées selon les politiques PF, EFD et D2PQ. Nous avons comparé aussi les résultats avec ceux obtenus par l'ordonnancement PAPS. Nous avons constaté que la combinaison CoSenS et politique d'ordonnancement tel que PF, EDF ou D2PQ améliore les résultats du trafic événementiel tout en limitant la dégradation subie par le trafic périodique causée par ces politiques. Cependant, nous avons remarqué aussi qu'après la saturation de CoSenS, même si celle-ci intervient pour une charge nettement supérieure à celle d'IEEE 802.15.4, le trafic périodique perturbe le trafic événementiel. Nous avons conclu que l'implémentation d'un mécanisme de contrôle de congestion est nécessaire afin de pallier ce problème. Le deuxième scénario simulé nous a permis de voir la limite d'une politique d'ordonnancement tel que EDF ou sa version améliorée D2PQ. Une des perspectives à court terme sera d'améliorer la politique D2PQ en intégrant d'autres paramètres.

CoSenS et P-CoSenS ne prennent pas en compte d'une manière explicite le facteur énergie au niveau des routeurs. L'implémentation de ces deux protocoles dans une couche integicielle permet d'utiliser n'importe quel protocole MAC qui gère l'énergie et qui s'appuie sur CSMA/CA. Cependant, la plupart des protocoles MAC avec gestion d'énergie négligent le compromis énergie-débit (que nous développons en détail dans le chapitre suivant). Or, CoSenS se base sur l'estimation du trafic. Cela offre un moyen efficace pour optimiser la consommation d'énergie en fonction du trafic.

Chapitre 5

S-CoSenS : le compromis énergie-débit

Sommaire

Ce chapitre présente S-CoSenS, une extension de CoSenS qui traite le compromis entre la minimisation de la consommation d'énergie et la préservation d'un débit suffisant pour l'application fonctionnant sur le réseau.

Le chapitre est organisé comme suit. Dans la première section, nous détaillons d'abord les motivations qui nous ont menés à proposer S-CoSenS. Ensuite, nous décrivons le protocole proposé. Après, nous présentons l'évaluation de performance de notre solution. Enfin nous concluons le chapitre dans la dernière section.

5.1 Motivations

L'énergie est un facteur important dans les réseaux de capteurs sans fil. Elle peut être considérée comme une métrique de QdS [Karl 2004]. En effet, du point de vue applicatif, certaines situations du terrain surveillé nécessitent une plus grande précision de détection ainsi q'une rapidité dans la transmission de l'information sans se soucier du budget énergie. Cela veut dire qu'un nœud doit effectuer plus fréquemment des mesures et les envoyer vers le responsable de la surveillance. C'est le cas par exemple lorsqu'une situation alarmante est détectée, le responsable doit donc suivre de plus près l'évolution de la situation afin de prendre les meilleures décisions. Dans d'autres situations, un nœud n'a pas besoin de transmettre toutes les valeurs

captées, il se contente par exemple d'envoyer la moyenne toutes les n mesures réalisées. Il peut même décider d'effectuer une seule mesure chaque n périodes au lieu d'une mesure par période et l'envoyer au responsable de la surveillance. C'est le cas lorsqu'il y a peu ou pas de changement de l'état de l'environnement surveillé. Le nœud préserve donc son énergie ainsi que celle des nœuds intermédiaires. Par conséquent, selon l'état de l'environnement surveillé, le réseau peut se trouver dans l'un des deux états suivants : minimisation de la consommation d'énergie pour augmenter la durée de vie du réseau et maintien d'un débit élevé afin de suivre une situation alarmante de plus près et prendre ainsi les bonnes décisions.

Pour minimiser la consommation d'énergie, la plupart des protocoles MAC dans les réseaux de capteurs sans fil tentent de réduire au maximum le duty-cycle. Le duty-cycle est la fraction du temps pendant laquelle le système est dans l'état "actif". Dans notre cas, l'état actif correspond à l'état où la radio du capteur est activée. Cependant, ajouter des périodes de sommeil dégrade considérablement le débit du réseau. Or, comme mentionné dans le premier paragraphe, celui-ci varie fortement selon la situation. Ainsi, un mécanisme d'auto-adaptation des périodes de sommeil et d'activité doit être mis en place afin de pallier ce problème. Pour cela, nous proposons une deuxième extension de CoSenS, baptisée S-CoSenS, "Sleep CoSenS". S-CoSenS est composé de deux couches, MAC et réseau, qui travaillent conjointement afin d'optimiser les performances du protocole en termes de consommation d'énergie et de problèmes de routage. Dans la couche MAC, le modèle "collecter puis envoyer en burst" adopté dans CoSenS a été étendu ; une période de sommeil a été ajoutée. Le protocole de routage utilisé est une variante du protocole R2A, la version optimisée du RA de ZigBee.

5.2 Dormir, Collecter puis envoyer en burst

5.2.1 Description générale

Au niveau de la couche MAC, S-CoSenS s'inspire à la fois de S-MAC et de B-MAC. Pour rappel, S-MAC [Ye 2002] est un protocole basé sur la contention. Un nœud utilisant S-MAC enchaîne un cycle composé d'une partie active et d'une partie inactive. Durant la partie active, il écoute le canal, reçoit et envoie des données. Durant la partie inactive il désactive son récepteur pour préserver l'énergie. Les nœuds se synchronisent entre eux pour qu'ils se réveillent et se rendorment en même temps. Ainsi ils adoptent un rendez-vous commun. Quant à B-MAC [Polastre 2004], chaque nœud échantillonne le temps en intervalles réguliers. Le nœud est par défaut en mode inactif. Au début de chaque intervalle il s'active, allume sa radio et vérifie l'état du canal (en effectuant un CCA). S'il détecte une activité, il reste actif pendant le temps nécessaire pour recevoir le paquet. Sinon, il se rendort. Du coté de l'émetteur, chaque transmission d'un paquet est précédée par la transmission d'un préambule. La taille du préambule doit correspondre au moins à l'intervalle d'échantillonnage. De cette façon le récepteur est averti et reçoit le paquet de données. S-CoSenS pos-

sède les avantages de ces deux protocoles et fournit d'autres améliorations. En effet, il enchaîne un cycle divisé en une partie active et une partie inactive comme le fait S-MAC. Toutefois, pendant les périodes d'inactivité, le nœud se réveille de temps en temps, vérifie l'état du canal et reste actif s'il détecte une activité comme le fait B-MAC. Cette combinaison élimine la complexité due au protocole de synchronisation dans S-MAC. De plus, puisque la partie active est divisée elle aussi en une période de collecte et une période de transmission, un seul préambule est utilisé pour transmettre l'ensemble des paquets reçus pendant la période de collecte. Ainsi, nous améliorons le débit.

Comme dans CoSenS, nous considérons un réseau organisé selon une architecture deux-tiers. Il est structuré selon un arbre et utilise l'adressage en arbre proposé par le standard ZigBee. Nous détaillerons par la suite le fonctionnement du routeur et du nœud simple.

5.2.2 Couche MAC

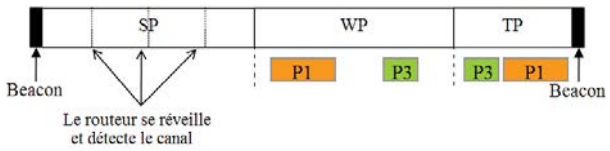
5.2.2.1 Fonctionnement du routeur

Lorsqu'un routeur rejoint le réseau après une exécution réussie de la procédure d'association (décrite dans le paragraphe 2.2.5.1), il commence à envoyer des beacons et ce d'une manière périodique. Un beacon est un message qui fournit des informations utiles destinées à ses voisins et qui servent à synchroniser les nœuds simples attachés à lui. Les routeurs utilisent une structure de super-trame définie comme étant la durée qui sépare deux beacons successifs. La super-trame est divisée en trois périodes ; période de sommeil (SP), période d'attente (WP) et période de transmission (TP). On appelle la somme de SP et de WP une sous-trame. Ainsi, pendant la SP, le routeur est inactif. Néanmoins, le routeur se réveille périodiquement pour détecter l'activité du canal. Lorsque la SP expire, le routeur se réveille et se met en mode collecte de données pendant la WP. A la fin de la WP, le routeur transmet tous les messages collectés en un seul burst. Finalement, le routeur transmet le beacon et passe au mode SP de nouveau. La structure de la super-trame est illustrée dans la figure 5.1(a).

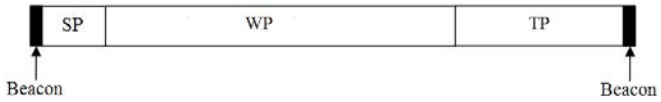
Aucune transformation n'a été opérée sur la WP. Elle est adaptée selon le trafic reçu en utilisant l'algorithme d'estimation par comparaison présenté dans le paragraphe 3.2.5.2. Par ailleurs, quelques modifications ont été apportées au fonctionnement de TP dans S-CoSenS par rapport à CoSenS. Elles seront détaillées dans le paragraphe suivant. La période de sommeil sera détaillée dans le paragraphe 5.2.2.3.

5.2.2.2 Modifications opérées sur la période de transmission

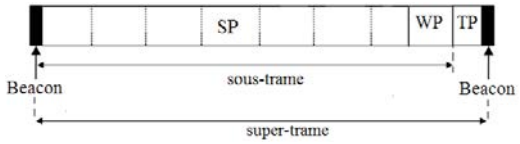
Les données collectées pendant la période de transmission sont transmises en un seul burst. Une première modification par rapport à CoSenS est qu'au début de cette période, le routeur envoie un préambule. Sa durée est supérieure à la durée



(a) Charge normale



(b) Charge élevée



(c) Charge faible

FIGURE 5.1 – Structure de la super-trame et taille des périodes de sommeil et d'attente en fonction de la charge.

de réveil périodique du routeur pendant la SP. De cette façon, tous les routeurs voisins qui sont inactifs vont détecter le canal comme étant occupé lorsqu'ils se réveillent et par conséquent restent actifs. Si le réseau est configuré pour qu'il reste actif tout le temps, la transmission du préambule est désactivée. A la fin de la transmission du préambule, le routeur transmet tous les paquets en file d'attente. Le protocole CSMA/CA est utilisé pour transmettre le préambule ou en cas d'échec de transmission d'un des paquets de données comme dans CoSenS. A la fin de la transmission des paquets de données, le routeur envoie le beacon et entre en mode sommeil.

Chaque routeur a au moins un message à transmettre pendant la TP qui est le beacon. S'il y a des paquets de données à transmettre dans la file d'attente, le beacon est transmis directement après la transmission de tous les paquets de données. Dans le cas contraire, le beacon est transmis en utilisant CSMA/CA et sans utilisation de préambule.

Notons qu'à l'expiration de la WP, le nœud ne peut démarrer la TP que si le canal de transmission est libre. Autrement dit, la WP est étendue jusqu'à ce qu'il le devienne. Ainsi, à chaque détection d'un canal occupé, la WP est prolongée d'une durée égale à d_S (décrite dans l'algorithme d'estimation par seuils, paragraphe 3.2.5.1)

5.2.2.3 Période de sommeil

Pendant la période de sommeil, le routeur désactive principalement sa radio pour préserver l'énergie. Toutefois, il l'active périodiquement pour détecter l'activité du

canal. S'il est libre, le routeur se rendort jusqu'au prochain instant de réveil ou à l'expiration de la SP. Dans le cas contraire, le routeur reste actif et reçoit les données du canal de transmission. Dans ce dernier cas, le routeur retourne en mode inactif lorsqu'il reçoit le beacon de l'émetteur, Sauf si la SP expire (il démarre la WP dans un tel cas). Si le beacon n'est pas reçu, le routeur retourne en mode inactif s'il ne reçoit pas de données pendant une durée d_S sauf si la SP expire.

La taille de la sous-trame, définie comme étant la somme de la SP et de la WP, est **constante** (on exclut les éventuels prolongements de la durée de la WP). Elle est définie a priori et peut être dimensionnée par rapport aux caractéristiques de l'application déployée (*e.g.* délai de bout-en-bout acceptable, durée de vie du réseau) [1]. Puisque la durée de la WP dépend de la quantité du trafic, celle de SP en dépend aussi. Ainsi, lorsque la charge du trafic augmente, la durée de la WP augmente et celle de la SP diminue. Inversement, lorsque la charge du trafic est faible, la durée de WP diminue et celle de la SP augmente. Ainsi, nous avons une adaptation dynamique entre les durées d'activité et d'inactivité dans le réseau. La figure 5.1 illustre les différentes dispositions d'une super-trame en fonction de la charge du trafic.

5.2.2.4 Fonctionnement du nœud simple

Un nœud simple est toujours en mode sommeil, autrement dit, sa radio est toujours désactivée, à moins qu'il ne reçoive un paquet provenant de la couche supérieure. Dans un tel cas, il se réveille et écoute le canal. Il continue à écouter le canal jusqu'à ce qu'il reçoive un beacon de la part d'un routeur voisin. Il extrait ensuite l'adresse du routeur qui l'a envoyé et les durées SP et WP (annoncées toujours dans le beacon). Le nœud simple se rendort aussitôt et ce pendant toute la période SP du routeur et se réveille à son expiration. Maintenant, le nœud simple est synchronisé avec la période d'attente du routeur ayant envoyé le beacon. Finalement, le nœud simple, enregistre l'adresse du routeur comme étant le prochain saut du paquet et le lui transmet en utilisant le protocole CSMA/CA.

Si le nœud a un autre paquet à transmettre ou si la transmission du paquet actuel a échoué, il doit vérifier s'il peut encore le transmettre pendant le temps restant avant l'expiration de la WP. Si tel est le cas, il tente de transmettre le paquet au routeur actuel. Sinon, il doit redémarrer toute la procédure décrite précédemment. Notons que le routeur choisi peut ne pas être le père du nœud lui-même car le premier routeur qui envoie son beacon après le réveil du nœud simple sera choisi.

1. Le dimensionnement se base sur les résultats de la simulation qui donnent la durée de vie du réseau et le délai de bout en bout en fonction de la charge du trafic et de la durée de la sous-trame

5.2.3 Couche réseau

5.2.3.1 Association

Notons par "racine", le routeur qui démarre le réseau et qui spécifie tous ses paramètres (comme la taille de la sous-trame et les paramètres d'adressage). La racine est équivalente au coordinateur ZigBee.

Le mécanisme d'association fonctionne comme suit. Lorsque la racine devient active, elle commence à envoyer des beacons périodiquement. Ainsi, elle commence à utiliser la structure de la super-trame directement. Si un routeur ou un nœud simple veut joindre le réseau, il commence par collecter des beacons pendant une durée déterminée, appelée période de collecte (PC), afin de construire la liste des routeurs voisins. A la fin de cette période, si la liste est vide, le processus recommence. Sinon, le nœud choisit, parmi la liste des routeurs voisins, celui le plus proche de la racine (cette information peut être extraite de l'adresse du routeur) et qui a le moins de fils (cette information est incluse dans le beacon). Le nœud reste actif jusqu'à ce qu'il reçoive le beacon suivant du routeur choisi. Maintenant il est synchronisé avec lui. Au démarrage de la période d'attente, il envoie sa requête d'association. A la réception de la requête, le routeur renvoie une réponse d'association immédiatement (*i.e.* sans attendre le début de la période de transmission). Si la réponse est positive, le nœud est maintenant associé au routeur choisi. Autrement, le nœud passe au routeur suivant dans la liste et répète la procédure de synchronisation puis d'envoie de requête d'association. Si tous les routeurs dans la liste sont sollicités sans succès d'association, le nœud recommence le processus depuis le début (en réexécutant la procédure de collecte de beacons). La procédure d'association est exécutée une seule fois pour toute la durée de vie du réseau, à moins que la racine ne décide de reconstruire le réseau.

Notons qu'à l'issue d'une association le père attribue à son fils une adresse. Puisque nous utilisons une version modifiée du protocole de routage en arbre de ZigBee, nous implémentons l'adressage hiérarchique ZigBee.

5.2.3.2 Protocole de routage

Nous utilisons le protocole de routage en arbre dans sa version améliorée (R2A) que nous avons décris dans le paragraphe 2.2.5.1. Nous avons ajouté deux modifications à R2A afin d'améliorer sa tolérance aux pannes. La première concerne le choix du prochain saut par un nœud simple. La deuxième concerne le choix du prochain saut par un routeur en cas de panne du routeur père. Dans ce qui suit, nous rappelons dans un premier temps R2A. Dans un deuxième temps, nous détaillons les deux améliorations ajoutées. Le nouveau protocole de routage est baptisé "routage en arbre modifié" (RAM).

Dans R2A, un nœud simple voulant transmettre un paquet à une destination donnée, envoie son message systématique à son routeur père. Celui-ci s'occupe du routage

vers la destination. Pour le routeur, si la destination est un de ses descendants, il calcule l'adresse du prochain saut et lui envoie le paquet. Dans le cas contraire (la destination n'est pas un de ses descendant), il vérifie si elle est un voisin ou un descendant d'un de ses voisins. Si c'est le cas, il lui envoie le paquet. S'il y a plusieurs voisins qui peuvent atteindre la destination, Il choisit celui qui a la profondeur la plus grande. Cette vérification des routeurs voisins constitue l'amélioration faite au protocole de routage en arbre de base (définie dans le standard ZigBee). Notons qu'elle ne nécessite aucun échange de données puisqu'on peut vérifier si un nœud est un descendant d'un autre juste en comparant leurs adresses. Finalement, si la destination n'est ni un descendant, ni un voisin, ni un descendant d'un des voisins alors le paquet est envoyé au père.

Dans la première amélioration, un nœud simple n'envoie pas systématiquement les paquets à son père. Au lieu de ça, le message est envoyé au routeur dont le beacon a été reçu le premier. Cela se fait de la façon suivante. Une fois qu'un message arrive à la couche réseau, celle-ci informe la couche MAC qui, à son tours, active la radio du nœud. Ensuite, elle attend la réception des beacons qui sont émis régulièrement par les routeurs. Le routeur ayant envoyé son beacon le premier après l'activation du nœud sera choisi comme prochain saut. Ainsi, le prochain saut est un routeur actif. Lors de la réception du beacon, la couche MAC du nœud en question informe la couche réseau de l'adresse de ce routeur. Le paquet est alors envoyé à la couche MAC pour continuer la procédure de transmission. Cette amélioration est illustrée dans la figure 5.2, $2^{ème}$ amélioration. Dans cet exemple, le nœud 8 a comme père le routeur 4. En se réveillant, le premier beacon reçu est celui envoyé par le routeur 3. Il choisit donc d'utiliser le routeur 3 comme prochain saut. Cette amélioration permet d'éviter deux problèmes. Premièrement, en choisissant le premier beacon reçu, le nœud élimine le coût d'attente inutile du beacon du père (énergie consommée). Deuxièmement, cela élimine le coût lié à l'indisponibilité du père que ce soit temporaire ou définitive. Le nœud peut ainsi relayer son message le plus rapidement possible sans perdre du temps pour découvrir l'indisponibilité du père (envoyer plusieurs fois le message sans réception d'acquittement ou non réception du beacon) et effectuer la procédure d'association.

Dans la deuxième amélioration, une modification a été introduite pour les routeurs afin d'améliorer leur tolérance aux pannes en utilisant la table des voisins. Celle-ci peut être construite efficacement en utilisant les beacons. Elle est accessible à la fois par la couche MAC par la couche réseau. Dans le cas où le paquet sera envoyé au père (*i.e.* la destination n'est ni un descendant, ni un voisin, ni un descendant d'un des voisins), une liste de prochains sauts possibles est extraite de la table des routeurs voisins. Un prochain saut est possible s'il n'a pas le même père que le routeur lui-même. La liste est ensuite ordonnée selon la profondeur du routeur ; celui qui a la profondeur la plus petite est placé en premier. Cette liste est aussi accessible par les couches MAC et réseau. Pendant la TP, si la transmission du paquet échoue après plusieurs tentatives de transmissions [2] et si la destination n'est pas le père lui-même

2. le nombre de tentatives est fixé à cinq dans les simulation

(la couche MAC connaît l'adresse de son père), le routeur remplace le prochain saut par l'adresse se trouvant dans la tête de la liste de prochains sauts possibles et tente de nouveau la transmission. Si la liste est vide, le message est supprimé. La $3^{ème}$ amélioration présentée dans la figure 5.2 illustre cette modification. Dans cet exemple, le nœud simple 10 envoie un paquet vers la racine 0. Lorsque le message arrive au routeur 5, il décide de l'envoyer à son père, le nœud 2. Il construit donc la liste de prochains sauts possibles, qui dans ce cas contient uniquement le routeur 4, et transmet le paquet au nœud 2. Seulement, celui-ci est indisponible. Après plusieurs tentatives, la couche MAC décide de remplacer l'adresse du nœud 2 par celle du nœud 4 et effectue la transmission de nouveau. La $3^{ème}$ modification est effective lorsque le prochain saut est le père. En effet, le puits est généralement la racine. De plus, les routeurs situés dans les premières profondeurs épuisent plus rapidement leurs batteries car la plupart du trafic passe par eux.

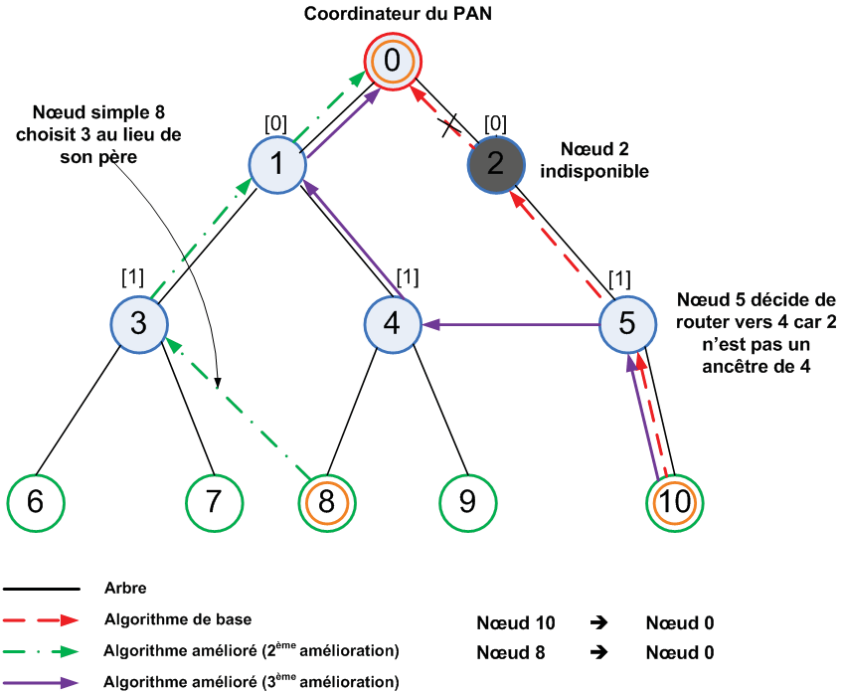


FIGURE 5.2 – Deux améliorations incluses dans le RAM.

5.3 Analyse des performances

5.3.1 Scénario

Nous considérons le réseau simulant un système de surveillance de patients présenté dans la section 3.3.4. La figure 5.3 reprend le réseau simulé. Afin d'étudier le protocole S-CoSenS, nous varions la taille de la sous-trame (SP+WP) et nous mesurons l'énergie moyenne consommée par routeur, le délai de bout-en-bout et le taux de

succès de transmission et ce pour trois différentes charges du trafic. L'analyse de la consommation d'énergie se fait uniquement pour les routeurs car les nœuds simples sont toujours en veille à moins qu'ils aient un paquet à transmettre et n'ont aucune influence sur la connectivité du réseau. Tous les nœuds envoient des données vers le ROOT, situé au centre du réseau (figure 3.15). Le tableau 5.1 résume les différents paramètres de simulation spécifiques à l'étude de S-CoSenS. Les autres paramètres sont fournis par le tableau 3.1.

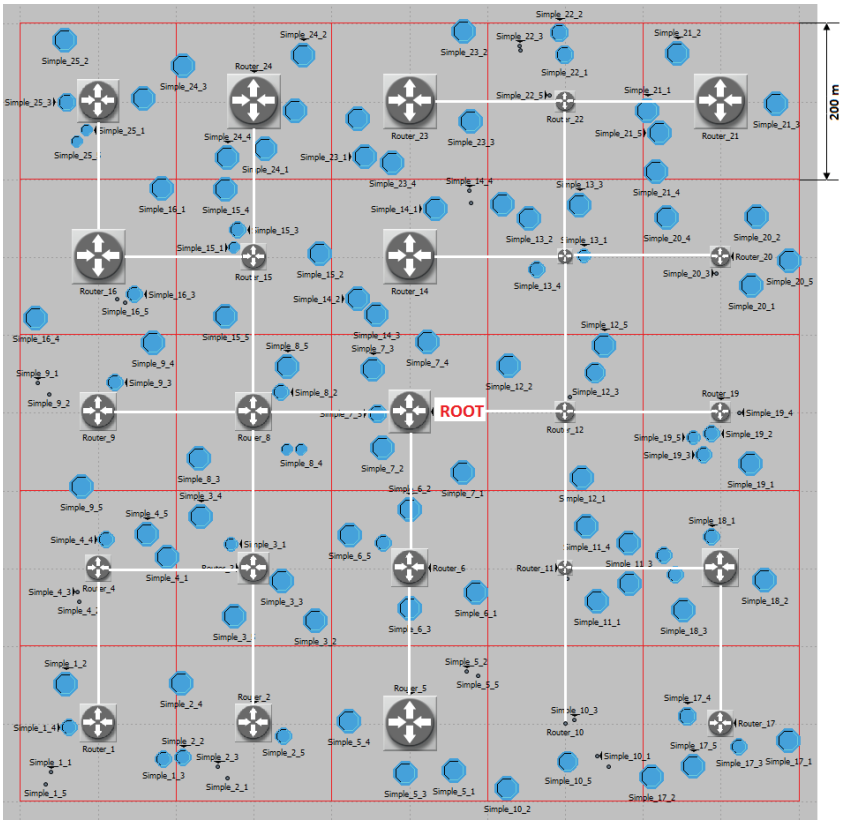


FIGURE 5.3 – Réseau multi-saut en grille.

TABLE 5.1 – Paramètres généraux de simulation spécifiques à l'étude de S-CoSenS.

Paramètre	**valeur**
Energie initiale (J)	1000
Durée de la simulation (min)	60
Temps d'inter-arrivée (s)	10, 60 et 120
Type de trafic	Poisson
Durée de la période de réveil (ms)	10

Pour le calcul de la consommation d'énergie, nous utilisons le modèle de consommation des capteurs micaz [Micaz 2011], dont les paramètres sont fournis par le

tableau 5.2.

Paramètre	valeur
Intensité du courant en mode réception (mA)	19.7
Intensité du courant en mode transmission (mA)	17.4
Intensité du courant en mode actif (mA)	8
Intensité du courant en mode inactif (μA)	15
Voltage (V)	3

TABLE 5.2 – Micaz mote parameters

5.3.2 Résultats

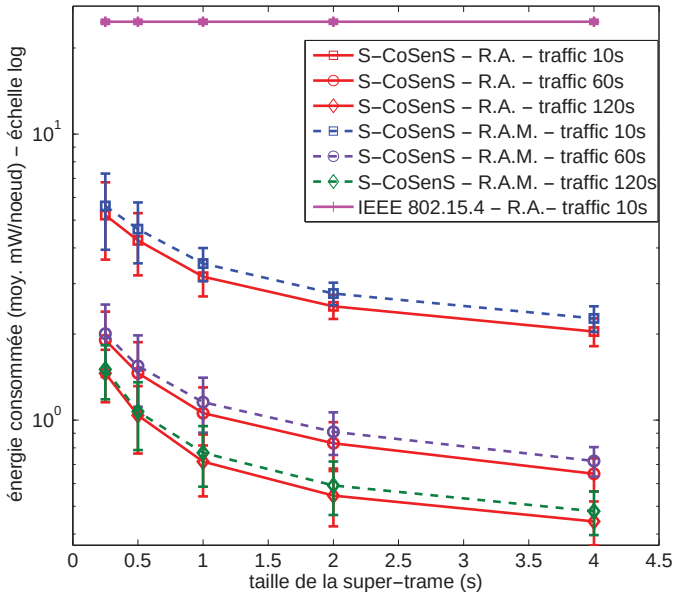


FIGURE 5.4 – Energie consommée par S-CoSenS pour différentes tailles de sous-trames et différents temps d'inter-arrivées. RA désigne "routage en arbre" et RAM désigne "routage en arbre modifié".

La figure 5.4 illustre l'énergie moyenne consommée par routeur en mW ainsi que l'écart type de cette consommation pour S-CoSenS et IEEE 802.15.4 utilisant la version non slottée de CSMA/CA. Nous remarquons tout d'abord que la consommation de IEEE 802.15.4 est largement supérieure à celle de S-CoSenS. Par ailleurs, elle est presque indépendante de la charge du trafic ; la différence entre les énergies consommées pour les différentes charges de trafic ne dépasse pas 0.8 mW. C'est pour cela que seulement la consommation d'énergie correspondant à un temps d'inter-

arrivée de 10 s est tracée. Dans CSMA/CA non slotté d'IEEE 802.15.4 la radio n'est jamais désactivée. C'est pour cela que la consommation d'IEEE est importante et est presque indépendante de la charge du trafic. Ensuite, pour S-CoSenS, la consommation d'énergie par routeur augmente lorsque la charge augmente. D'une part, les routeurs se réveillent plus souvent pendant la durée de la SP pour récupérer les données envoyées par les routeurs voisins. D'autre part, la durée de la WP est plus longue puisque elle est adaptée en fonction de la charge du trafic. Puis, la consommation d'énergie par routeur diminue lorsque la taille de la sous-trame augmente. En effet, l'augmentation de la taille de la sous-trame implique l'augmentation de la taille de la SP pour une charge donnée, ce qui diminue la consommation d'énergie. De plus, nous observons que la consommation est plus uniforme lorsque la taille de la sous-trame est longue. Les routeurs situés près du ROOT consomment plus d'énergie car toutes les données passent par ces routeurs pour atteindre le ROOT. Pour des durées de sous-trames faibles, la proportion de l'état actif, composé de WP et des réveils pendant les SP, est importante pour les routeurs situés près du ROOT et est beaucoup plus faible pour ceux loin du ROOT. L'effet sur la consommation d'énergie est donc plus remarquable. Lorsque la taille de la sous-trame augmente, la proportion de l'état actif pour tous les routeurs est beaucoup plus petite que la taille de la SP (pour rappel la durée de la WP est limitée à 70 ms dans les simulations). Son effet sur la consommation d'énergie est donc moins remarquable. Finalement, la consommation d'énergie de S-CoSenS utilisant le routage en arbre modifié est légèrement supérieure à celle obtenue par le RA car il effectue une tentative de transmission supplémentaire avant la suppression du paquet, ce qui consomme un peu plus d'énergie. Cependant, cette légère augmentation est justifiée vu les taux de succès de transmission obtenus par l'algorithme de routage modifié (voir le dernier paragraphe de cette sous-section pour l'argumentation de cette constatation).

La consommation moyenne par routeur nous permet de déterminer la durée de vie moyenne du réseau. L'équation 5.1 donne la durée de vie moyenne du réseau en fonction de la capacité de la batterie (C_{batt}) et de la consommation moyenne d'énergie par routeur.

$$Durée\ de\ vie\ moyenne(s) = \frac{C_{batt}\ (mAh) \cdot V \cdot 3600}{Consommation\ moyenne\ d'énergie\ (mW)} \tag{5.1}$$

Par exemple, la capacité typique d'une pile AA type alkaline, utilisée par les capteurs micaz, est de 2500 mAh. Pour une charge égale à un message chaque 60 s et une taille de sous-trame égale à 2 s, S-CoSenS avec RAM consomme en moyenne 0.92 mW, ce qui donne une durée de vie du réseau de 339 jours.

La figure 5.5 illustre les délais de bout-en-bout obtenus par S-CoSenS pour différentes tailles de sous-trames ainsi que par IEEE 802.15.4. Ce dernier obtient toujours de meilleurs résultats car les routeurs sont tout le temps actifs. Le délai de bout-en-bout augmente lorsque la taille de la sous-trame augmente. En effet, le temps de service est plus long car il faut attendre la TP pour transmettre les paquets, ce qui

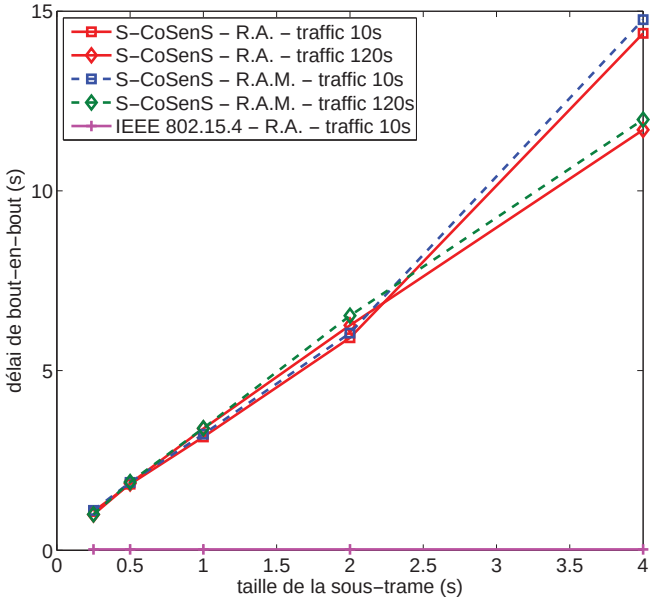


FIGURE 5.5 – Délai de bout-en-bout de S-CoSenS pour différentes tailles de sous-trames et différents temps d'inter-arrivées. RA désigne "routage en arbre" et RAM désigne "routage en arbre modifié".

prend plus de temps lorsque la durée de la sous-trame est longue.

La figure 5.6 montre les taux de succès de transmission obtenus par S-CoSenS et IEEE 802.15.4. Les performances de S-CoSenS sont plus faibles que celles de IEEE 802.15.4. En effet, le débit offert par IEEE 802.15.4 est plus important que celui de S-CoSenS car le premier est tout le temps actif. Cela engendre des collisions plus fréquentes et donc un taux de succès de transmission plus petit dans le cas de S-CoSenS. L'utilisation de RAM permet de limiter ce problème. En effet, effectuer une tentative de transmission de plus avec le choix d'un autre prochain saut lors de l'échec de transmission vers un routeur donné, permet d'améliorer le taux de succès de transmission sans augmenter excessivement la consommation d'énergie. Nous remarquons que le RAM permet une amélioration de taux de succès de transmission comprise entre 10,12% et 18.70% avec une augmentation de la consommation d'énergie comprise entre 3.02% et 7.64%.

5.4 Analyse du pire délai de bout-en-bout

Considérons le réseau illustré dans la figure 5.7. Le nœud 4 qui est un nœud simple envoie un paquet vers le nœud 0. Le nombre de sauts est égal à quatre. La figure illustre le pire délai de bout-en-bout. Il correspond au cas où le nœud 4 se réveille juste après la transmission du beacon par le nœud 3 et un déphasage maximal

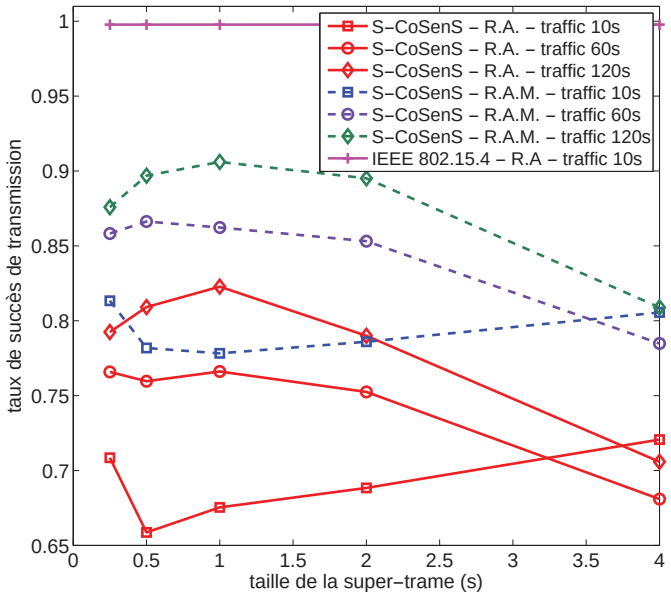


FIGURE 5.6 – Taux de succès de transmission de S-CoSenS pour différentes tailles de sous-trames et différents temps d'inter-arrivées. RA désigne "routage en arbre" et RAM désigne "routage en arbre modifié".

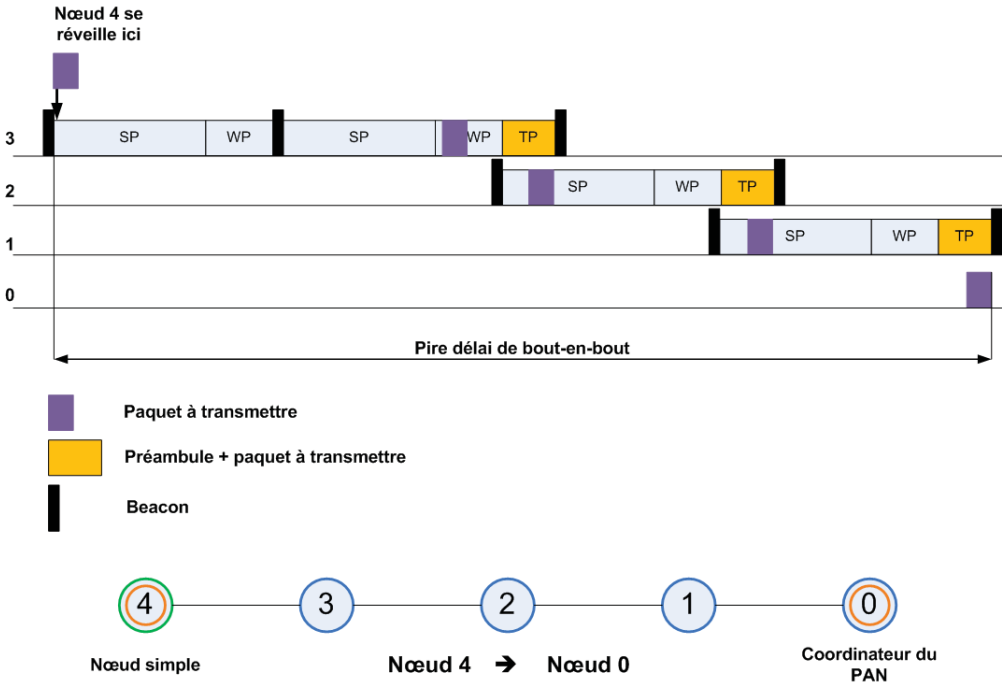


FIGURE 5.7 – Pire délai de bout-en-bout.

entre les supertrames des nœuds intermédiaires. Le déphasage est maximal lorsque le beacon d'un nœud est transmis juste avant la fin de la WP du nœud précédent. Le pire délai de bout-en-bout d'un paquet p pour ce réseau est égal à :

$$d_p = 4 \cdot (WP + SP) + T_{tx}(beacon) + T_{tx}(Preambule) + T_{tx}(paquet) \tag{5.2}$$

Dans l'équation 5.2, $T_{tx}(i)$ correspond au délai de transmission de i. $WP + SP$ est égale à la taille de la sous-trame, notée T_{st}, qui est constante. Le pire délai de bout-en-bout d'un paquet p généré par un nœud simple qui se trouve à n sauts de la destination est donné par l'équation 5.3.

$$d_p(n) = n \cdot T_{st} + T_{tx}(beacon) + T_{tx}(Preambule) + T_{tx}(paquet) \tag{5.3}$$

5.5 Conclusion

Dans ce chapitre nous avons proposé une deuxième extension à CoSenS que nous avons appelée S-CoSenS pour "Sleep CoSenS". S-CoSenS est composé d'une couche MAC qui ajoute une période de sommeil aux périodes d'attente et de transmission et d'une couche réseau intégrant une version modifiée du protocole de RA de ZigBee. Actuellement, nous somme en train de porter S-CoSenS sur Contiki. Les travaux sont effectués par un stagiaire recruté à cet effet.

Nous remarquons qu'en travaillant d'une manière conjointe, une pile protocolaire peut offrir des garanties de QdS avec des coûts faibles. La modification apportée au protocole de routage en arbre pour fournir une alternative de prochain saut à la couche MAC, est un exemple de collaboration inter-couches, qui nous a permis d'améliorer le taux de succès de transmission avec un coût énergétique peu élevé. Ainsi, faut-il réfléchir à une nouvelle architecture permettant à toutes les couches de travailler d'une manière coordonnée afin d'améliorer les performances du réseau et de garantir un niveau suffisant de QdS avec les coûts les plus faibles possibles ? D'autant plus que fournir la QdS dans un réseau implique toutes les couches de la pile protocolaire.

Chapitre 6

Conclusion et perspectives

6.1 Conclusion générale

Petit à petit, les réseaux de capteurs sans fil commencent à pénétrer dans notre vie quotidienne. Des applications diverses et variées, basées sur ces réseaux, se développent pour améliorer notre confort et sécurité. Les capteurs sans fil autonomes offrent un avantage de coût et de facilité de déploiement puisqu'ils embarquent un système de communication sans fil et sont souvent alimentés par une batterie. Cependant, cet avantage impose des contraintes fortes en ce qui concernent les protocoles développés. En effet, ils doivent prendre en compte la consommation d'énergie, les pannes que ce soit temporaires ou définitives et la facilité de passage à l'échelle. A toutes ces contraintes s'ajoutent celles sur les trafics générés par des applications comme par exemple le délai, le débit, la fiabilité, *etc.* La coexistence de plusieurs types de trafics dans un réseau de débit faible crée un besoin de différenciation de service afin que les données les plus importantes puissent être traitées avec priorité. Malheureusement rares sont les protocoles existants qui supportent la QdS.

Dans cette thèse, nous avons proposé un ensemble de solutions permettant la gestion de la QdS sur des réseaux de capteurs basé sur le protocole CSMA/CA (le cas de la plupart des réseaux existants).

Dans le chapitre 2, nous avons passé en revue les principales solutions existantes en termes de protocoles MAC et routage développés pour les réseaux de capteurs sans fil. Premier constat : le nombre de protocoles sur papier est très important mais rare sont ceux vraiment adoptés et implémentés réellement. On notera l'adoption large du standard IEEE 802.15.4 ; la plupart des réseaux implémente au moins CSMA/CA non-slotté. Sans surprise, la majorité des protocoles MAC se focalisent sur la minimisation de la consommation d'énergie. Par contre rares sont ceux qui se préoccupent aussi de la QdS. Quant aux quelques solutions gérant la QdS, elles se basent généralement sur TDMA ou CSMA/CA. Si cette solution TDMA est convenable pour des réseaux de petite taille organisés en étoile et déployés dans des environnements connus, sa généralisation se heurte vite au problème de passage à échelle car la synchronisation des nœuds et l'ordonnancement des "time slots" deviennent coûteux. A tout cela s'ajoute le problème de la robustesse vis-à-vis de la dynamique du réseau (variation de charge, de topologie, etc.) car la réservation statique des "time slots" ne permet pas de s'adapter aux changements dynamiques et fréquents du réseau. Pour les solutions basées sur CSMA/CA, ils souffrent du problème de dégradation des performances lorsque la charge du trafic augmente.

Ces constats nous ont poussés à développer une autre approche : ajouter des mécanismes de la gestion de QdS au-dessus du protocole CSMA/CA non-slotté. Cette approche se veut premièrement pragmatique car facilement acceptable/implémentable du fait que la plupart des réseaux fonctionne déjà avec CSMA/CA qui, en plus, supporte bien le passage à l'échelle. Elle se veut aussi performante en termes de paramètres classiques tels que le débit, le délai et l'efficacité énergétique. Enfin, elle se veut auto-adaptative aux conditions changeantes du réseau.

Pour y parvenir, le premier problème que nous avons résolu est le problème de mauvaise performance de CSMA/CA lors d'une charge élevée momentanée (rafale de trafic). Cette situation peut se produire couramment dans un réseau dense où un événement est détecté simultanément par plusieurs capteurs qui transmettent tous des données vers un nœud routeur. Le chapitre 3 détaille CoSenS, un protocole d'accès basé sur CSMA/CA permettant d'améliorer ses performances en termes de délai, de débit et de taux de succès de transmission. Le réseau est structuré en deux niveaux : nœuds simples et routeurs. Le fonctionnement de CoSenS qui se base sur deux périodes "collecter des données des nœuds simples ou des routeurs voisins" puis "envoyer en rafale des paquets collectés". Les deux périodes s'auto-ajustent pour s'adapter aux conditions de trafic grâce à un algorithme d'adaptation simple qui estime en ligne la situation courante du trafic réseau. Une des propriétés remarquables de l'algorithme proposé est l'auto-adaptation des périodes de transmission en rafale des routeurs voisins, qui évite la superposition de ces périodes, réduisant ainsi les collisions. La séparation de la période de collecte avec la période de transmission en rafale permet d'éviter que le trafic entrant entre en collision avec le trafic sortant (segmentation du trafic du même domaine de collision radio par deux périodes temporelles). La période de collecte met en file d'attente les paquets entrants, résolvant ainsi un problème difficile qui est l'ordonnancement des paquets entrants dans un nœud (routeur), ce qui donne la possibilité d'ordonnancer différemment les paquets selon leur priorité.

CoSenS constitue la contribution principale de la thèse. Deux extensions de CoSenS ont été développées. La première, détaillée dans le chapitre 4, concerne la différentiation de service. Il s'agit d'introduire la notion de priorité (P-CoSenS) et d'implémenter quelques politiques d'ordonnancement tel que EDF, D2PQ ou PF. Nous avons constaté que cette extension améliore les résultats du trafic événementiel (supposé prioritaire) tout en limitant la dégradation subie par le trafic périodique (non prioritaire) causée par ces politiques. La deuxième extension décrite dans le chapitre 5 intègre la dimension énergie en ajoutant une période de sommeil à CoSenS. Cette extension, baptisée S-CoSenS, possède deux caractéristiques. La première concerne l'auto-adaptation de la période d'activité et de sommeil en fonction de la charge de trafic du routeur. La deuxième concerne l'optimisation conjointe du protocole MAC et routage afin d'améliorer la tolérance aux pannes du réseau.

6.2 Perspectives

6.2.1 Perspectives à court terme

Nous avons montré dans le chapitre 4 les limites de la politique d'ordonnancement de type EDF ou sa version améliorée D2PQ. Ainsi la première perspective à court terme consiste à améliorer les décisions d'ordonnancement en intégrant d'autres paramètres. L'objectif reste de trouver la meilleure équation de calcul de la priorité permettant d'effectuer un ordonnancement global des paquets circulant dans le réseau en se basant uniquement sur des décisions locales effectuées au niveau de chaque nœud.

La deuxième perspective à court terme concerne S-CoSenS que nous somme en train de porter sur Contiki. Il s'agit dans un premier temps de comparer ses performances avec des protocoles qui gèrent l'énergie telle que B-MAC ou X-MAC. Dans un deuxième temps, nous étudions les possibilités d'amélioration de notre solution que ce soit au niveau MAC en optimisant encore plus la consommation d'énergie ou au niveau du routage en introduisant le paramètre énergie dans les décisions de routage afin de répartir uniformément la consommation d'énergie dans le réseau.

6.2.2 Perspectives à long terme

La QdS adressée dans cette thèse est du " best-effort " sans garantie. La différentiation de service seule n'est pas suffisante pour offrir de la QdS garantie. Il faut ajouter à cela d'autres mécanismes. Le plus important d'entre eux au niveau MAC étant le contrôle de congestion. On envisage uniquement l'approche réactive car celle préventive, faisant intervenir des mécanismes de contrôle d'admission, de lissage et de contrôle du trafic, serait trop coûteuse pour les réseaux de capteurs sans fil avec des ressources extrêmement limitées. Ainsi, l'idée de permettre aux routeurs d'agir sur les données qui circulent dans le réseau en fonction de son état de congestion serait intéressante à explorer. Dans les réseaux de capteurs sans fil, la notion de congestion inclut la notion de la charge du réseau mais peut aussi intégrer le facteur énergie dans le sens où un nœud n'a plus assez d'énergie pour transmettre ou recevoir tous les paquets qui sont supposés circuler à travers lui. Deux approches sont possibles en cas de congestion ; la suppression de paquets et l'agrégation de données. Quand à la suppression de données, le concept de la dégradation contrôlée selon le modèle (m,k)-firm développée au sein de l'équipe TRIO semble une approche intéressante et bien adaptée aux réseaux de capteurs sans fil [Li 2007]. En effet, la suppression de paquets d'une façon aléatoire peut mener à la situation où on n'a aucune information sur l'état du système pendant une période donnée parce qu'on a supprimé toutes les données provenant d'un des capteurs pendant cette période. La dégradation contrôlée permet de supprimer des paquets d'une manière plus intelligente de telle manière qu'on dégrade la précision des mesures (un paquet toutes les cinq minutes au lieu d'un toutes les minutes) sans pour autant créer la situation non souhaitable où aucune information n'est disponible. Dans la deuxième approche, le

routeur décide d'agréger les données avant de les envoyer. Un exemple d'agrégation consiste à calculer la moyenne des cinq dernières mesures d'un capteur et ainsi envoyer en un seul message. De la même façon, le routeur peut calculer la moyenne des valeurs reçues par les différents capteurs situés dans une même zone et envoyer uniquement le résultat. Par ailleurs, dans la couche réseau le routage multi-chemins est un mécanisme nécessaire pour garantir certains paramètres tel que la fiabilité de transmission et la tolérance aux pannes. On peut envisager qu'en fonction de la nature du service demandé par un paquet, il peut être transféré en utilisant un ou plusieurs chemins. Que ce soit dans la couche réseau ou MAC, l'application de ces mécanismes pose les problèmes suivants. En supposant qu'un routeur a tous les paquets à relayer, quels sont ceux à supprimer, à agréger, à transmettre avec plus de fiabilité, *etc.* ? De plus, comment faire si par exemple la couche réseau décide d'envoyer le même paquet sur plusieurs chemins alors que la couche MAC décide de les agréger pour diminuer la charge ? Par conséquent, Il est nécessaire de coordonner le fonctionnement des couches. Nous avons remarqué qu'en travaillant d'une manière conjointe, une pile protocolaire peut améliorer l'offre de la QdS. Cependant, on pense que ce n'est pas suffisant et qu'on a besoin de redéfinir l'architecture protocolaire pour permettre à toutes les couches de travailler d'une manière conjointe afin d'améliorer les performances du réseau et de garantir un niveau suffisant de QdS avec les coûts les plus faibles possibles.

Les perspectives se résument alors aux trois points suivant.

- Concevoir des mécanismes de contrôle de congestion. On envisage les deux approches ; celle de dégradation contrôlée selon le modèle (m,k)-firm et celle d'agrégation des données.
- Concevoir des protocoles de routage multi-chemins pour l'amélioration de la fiabilité de transmission et de tolérance aux pannes.
- Concevoir une architecture protocolaire permettant premièrement l'amélioration des performances du réseau en terme de durée de vie, de fiabilité, de délai et de débit et deuxièmement la garantie d'une qualité de service suffisante.

Annexe A

Définitions

ad-hoc : un réseau ad-hoc est un réseau sans fil capable de s'organiser sans infrastructure définie préalablement (source Wikipedia).

Beacon : un beacon est un terme anglais qui désigne balise. Il s'agit d'un message envoyé périodiquement ou pseudo-périodiquement pour synchroniser des nœuds ou leur délivrer des informations.

Burst : un burst de trafic désigne un ensemble de données transférées sans interruption d'un nœud à un autre.

CCA : CCA est acronyme anglais qui désigne une procédure de la couche physique permettant la vérification de l'état du canal.

Charge de saturation : c'est la charge maximale au delà de laquelle le débit n'augmente plus.

CW : CW est un acronyme anglais qui désigne "fenêtre de contention". Le terme est utilisé dans le protocole CSMA/CA mais est défini différemment selon la variante du protocole. Dans IEEE 802.11, il désigne la borne supérieure du back-off. Dans CSMA/CA slotté de IEEE 802.15.4, il désigne le nombre de fois où le canal est détecté libre (idle) avant d'accéder au canal.

Détection virtuelle de la porteuse : ce mécanisme est lié au protocole RTS/CTS de IEEE 802.11. Les paquets RTS et CTS incluent la durée de la transaction entre les deux nœuds ayant échangé ces messages. Cette information permet aux nœuds voisins de prévoir l'état d'occupation du support physique.

GPS : c'est un acronyme anglais qui désigne "système de positionnement mondial" est un système de géo localisation fonctionnant au niveau mondial.

Préambule : c'est un message spécial, contenant ou pas des informations utiles et permettant d'annoncer aux récepteurs possibles la transmission d'un message utile.

Qualité de service : la recommandation E.800 (09/2008) de l'union internationale des télécommunications (UIT) a définie la QdS comme étant l'"Ensemble des caractéristiques d'un service de télécommunication qui lui permettent de satisfaire aux besoins explicites et aux besoins implicites de l'utilisateur du service"

Rapport cyclique : le rapport cyclique désigne, pour un phénomène périodique, le ratio entre la durée du phénomène sur une période et la durée de cette même période.

Système temps réel dur : on parle de temps réel dur quand les événements traités trop tardivement ou perdus provoquent des conséquences catastrophiques pour la bonne marche du système (perte d'informations cruciales, plantage, *etc*). Les systèmes à contraintes dures ne tolèrent qu'une gestion stricte et bornée du temps afin de conserver l'intégrité du service rendu et sont toujours respectés.

Système temps réel mou : on parle de temps réel mou quand les événements traités trop tardivement ou perdus sont sans conséquence catastrophique pour la bonne marche du système. Un système temps réel mou tolère le dépassement des contraintes temporelles dans certaines limites au-delà desquelles il devient inutilisable.

TCP/IP : c'est la suite des protocoles utilisés pour le transfert des données sur Internet.

TDMA : est un acronyme anglais qui désigne l'accès multiple à répartition dans le temps. C'est une technique de multiplexage temporel permettant de transmettre plusieurs signaux sur un le même canal.

Annexe B

Modèles de simulation sous OPNET Modeler

B.1 Le simulateur OPNET Modeler

OPNET Modeler, nommé OPNET par la suite, est un simulateur réseau à événements discrets. Outre la bibliothèque de protocoles fournis, OPNET Modeler inclut un environnement de développement permettant la conception de nouveaux modèles. Le simulateur contient aussi des outils de collecte et d'analyse de données.

L'environnement OPNET est organisé d'une manière hiérarchique avec un éditeur pour chaque niveau (figure. B.1). On distingue alors les éditeurs suivants.

- **Éditeur du projet :** il permet de définir la topologie du réseau, de configurer le réseau en spécifiant les paramètres de chaque nœud et de spécifier les paramètres de simulation tel que la durée et les statistiques à collecter. Il inclut aussi les outils d'analyse des données collectées.
- **Éditeur de nœud :** chaque nœud est modélisé par un ensemble de modules interconnectés. L'éditeur de nœud permet de spécifier les différents modules d'un nœud et les liens entre eux. On distingue quatre types de modules : les émetteurs, les récepteurs, le processeur et la queue. Le comportement des émetteurs et des récepteurs est déjà prédéfini. Ils ne peuvent être configurés qu'à travers leurs paramètres. Le comportement du processeur et de la queue est défini par un modèle de processus présenté dans le point suivant.
- **Éditeur de processus :** un processus représente une machine à état finis définissant un comportement particulier, tel qu'une couche MAC ou réseau par exemple. L'éditeur de processus permet de créer cette machine à état finis. Chaque état contient le code, écrit en langage C, des tâches à exécuter à l'entrée et à la sortie. La transition spécifie la condition pour passer d'un état à un autre. Un processus peut faire appel à un autre. On parle alors de processus fils. Les processus sont aussi organisés d'une manière hiérarchique. Par exemple un processus MAC peut faire appel à un processus CSMA/CA.

B.2 CoSenS et P-CoSenS

Le modèle d'un nœud CoSenS est illustré par la figure B.2. Il est composé de six modules formant les couches application, réseau, MAC et physique. Notons qu'il

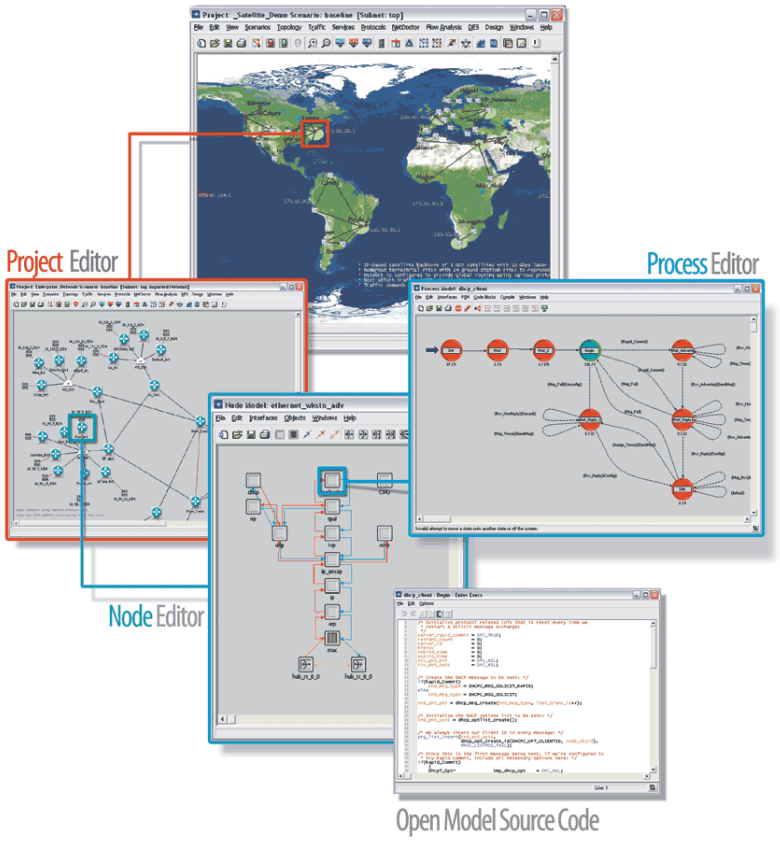


FIGURE B.1 – L'environnement OPNET [OPNET Technologies].

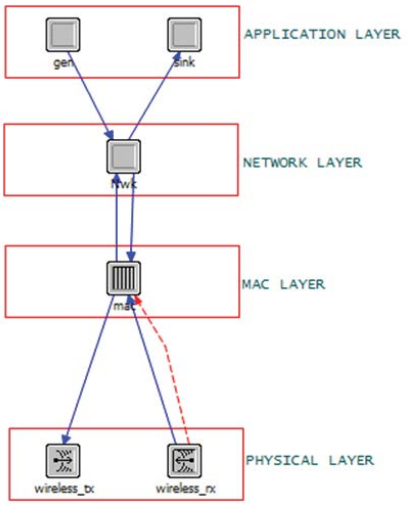
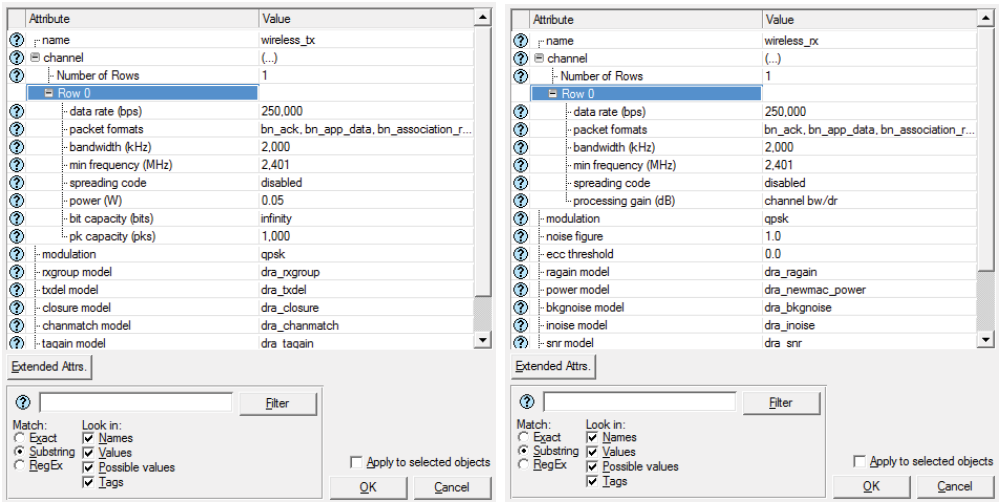


FIGURE B.2 – Modèle OPNET d'un nœud CoSenS.

y a deux modèles de nœuds CoSenS ; routeur et nœud simple. Cependant ils sont identiques et utilisent les mêmes modèles de processus. La différence est dans les valeurs des paramètres. Par exemple, la structure CoSenS est activée pour le routeur et désactivée pour le nœud simple.

B.2.1 Couche physique

La couche physique est composée de deux modules ; "wireless_tx" pour la transmission et "wireless_rx" pour la réception. La figure B.3 détaille les paramètres de l'émetteur et du récepteur. On considère une radio émettant sur une fréquence 2.402 GHz sur une bande de 2 KHz, ayant un débit de 250 kb/s et utilisant la modulation QPSK.



(a) Émetteur. (b) Récepteur.

FIGURE B.3 – Fenêtres des paramètres.

B.2.2 Couche MAC

Le modèle de processus de la couche MAC est représenté par la figure B.4. Il est composé de deux états. Le premier, nommé "init", permet d'initialiser tous les paramètres du modèle. Le deuxième représente l'état actif et dans lequel toutes les opérations du protocole MAC sont réalisées. L'état actif contient sept transitions. Chacune des transitions gère un événement particulier. La transition est composée d'une condition et d'une fonction illustrées de la façon suivante ; $(condition)/fonction(params)$. Si une condition est vraie, la transition est franchie et la fonction correspondante est exécutée. Notons que l'éditeur de processus inclut une fenêtre permettant de définir des fonctions en C. La liste des conditions est donnée ci-après.

– PHY_RECEPTION signale une réception d'un paquet de la couche physique.
– POWER_THRESH_STAT signale la puissance de réception du dernier paquet reçu par la couche physique.
– TX_DONE signale un succès de transmission d'un paquet par le protocole CSMA/CA.
– TX_FAILED signale un échec de transmission d'un paquet par le protocole CSMA/CA.
– SEND_TIMER signale l'expiration de la période d'attente.
– ACK_TIMER signale l'expiration du timer d'acquittement.
– NWK_RECEPTION signale la réception d'un paquet de la couche réseau.

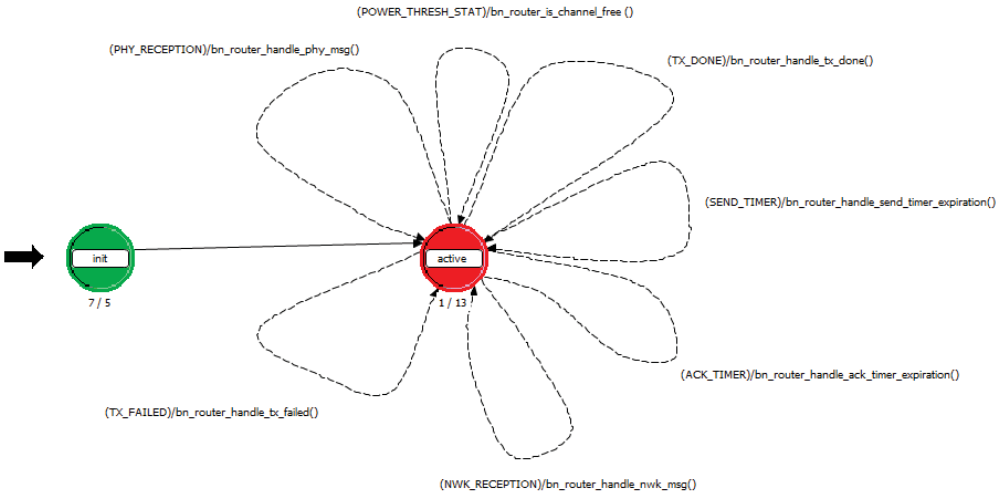


FIGURE B.4 – Modèle de processus MAC.

Si la fonctionnalité de CoSenS est activée, chaque paquet reçu de la part de la couche réseau est systématiquement mis dans la file d'attente. De plus, si l'une des politiques d'ordonnancement est activée, la priorité du paquet est calculée et le paquet est inséré dans la file selon cette priorité. A l'expiration de la période d'attente et si la file d'attente n'est pas vide, le premier paquet est extrait et le protocole CSMA/CA est invoqué pour l'envoyer. Si ce dernier renvoie un succès de transmission, TX_DONE, le timer d'acquittement est armé. Lorsque l'acquittement est reçu de la part de la couche physique, le timer est annulé et le le prochain paquet est extrait de la file d'attente. Celui-ci sera transmis directement sans invocation de CSMA/CA. Lorsque la file se vide le timer de la période d'attente, correspondant à l'événement SEND_TIMER est armé de nouveau. Lors d'un échec de transmission par CSMA/CA, TX_FAILED, ou à l'expiration du timer d'acquittement, ACK_TIMER, le protocole CSMA/CA sera invoqué systématiquement pour le prochain paquet à transmettre (que ce soit le paquet lui même ou le prochain dans la file d'attente).

CSMA/CA est implémenté comme processus fils. Il est illustré par la figure B.5. Le processus démarre lorsqu'un premier appel est effectué. Dans ce cas, il entre dans l'état *init* dans lequel il initialise ses variables pour la première fois et arme le timer

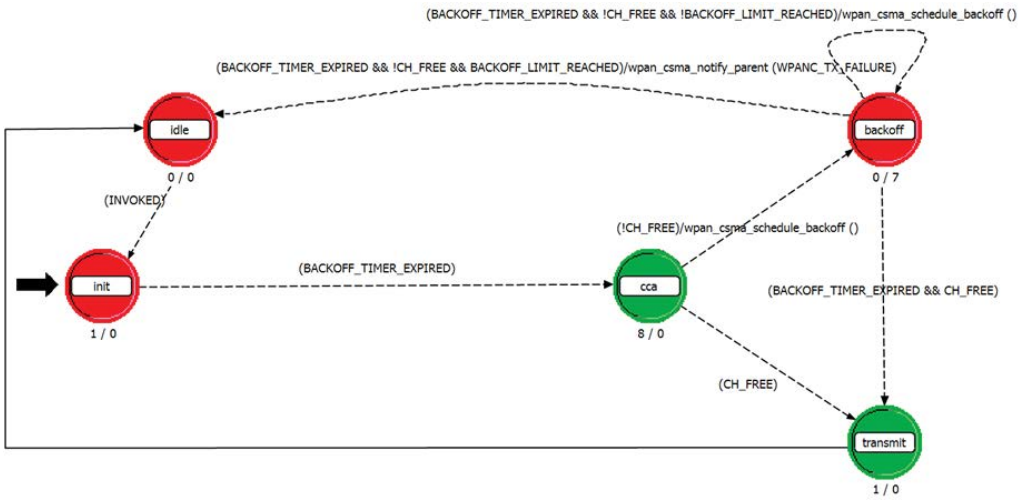


FIGURE B.5 – Modèle de processus du protocole CSMA/CA.

de backoff. A l'expiration du timer, BACKOFF_TIMER_EXPIRED, le processus teste dans l'état *cca* si le canal est libre. Dans le cas d'une réponse positive, il envoie le paquet, annonce le succès de transmission au processus MAC et passe à l'état *idle*, état d'attente. Dans le cas contraire, il passe à l'état backoff et arme a nouveau le timer de back-off pour une nouvelle tentative. Si le nombre maximal de tentatives est atteint et le paquet n'est pas transmis, il déclare un échec de transmission à la couche MAC et passe au mode *idle*. Si le processus CSMA/CA est invoqué de nouveau, il passe à l'état *init* de nouveau. Cependant, il n'initialisera pas de nouveau tous les paramètres, il récupère juste le paquet et arme le timer de backoff

B.2.3 Couche réseau

La couche réseau comporte un seul module. Le modèle de processus de ce module la est représenté par la figure B.6. Après l'initialisation des paramètres dans l'état *init*, le processus passe par l'état association si le routage en arbre est activé ou directement à l'état attente d'un événement, *idle* si le routage statique est utilisé. Dans le dernier cas, la table de routage est déjà remplie dans l'état *init*. L'état *idle* contient trois transitions correspondant à la réception d'un paquet de la couche MAC, MAC_RECEPTION, la réception d'un paquet de la couche application, APL_RECEPTION, et d'un état dit par défaut, default, dans lequel on récupère tout événement ne correspondant pas aux deux premiers.

La procédure d'association est implémentée comme un processus fils du processus réseau. Il est invoqué dans l'état *association*. La figure B.7 montre le dit processus. Après l'initialisation des paramètres, le processus passe à l'état $father\_choice$ dans lequel le nœud récupère la liste des nœuds voisins découverts, sélectionne le meilleur choix et lui envoie une demande d'association. Si la liste est vide, le processus envoie un message *Hello* puis passe à l'état $wait\_for\_neig$ dans lequel le nœud écoute le

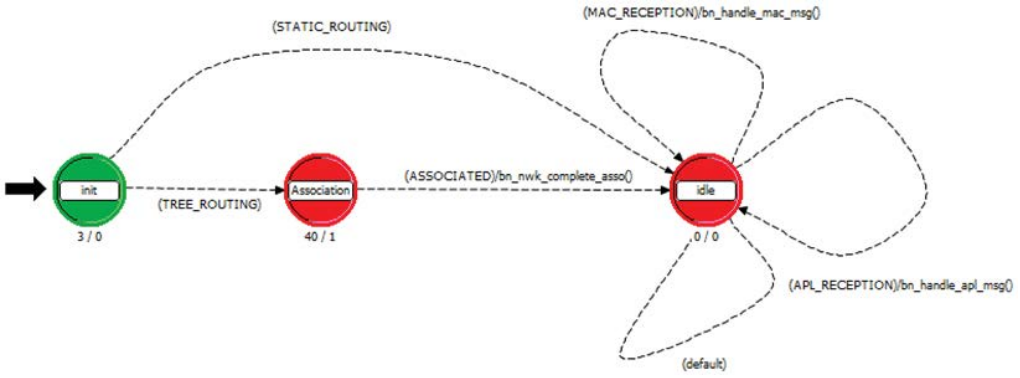


FIGURE B.6 – Modèle de processus de la couche réseau.

canal pour collecter les réponses au message *Hello* des routeurs voisins et ce pendant un certain temps. A l'expiration de ce temps, le processus passe à nouveau à l'état $father\_choice$ pour choisir le meilleur père. Notons que la liste des routeurs voisins se fait par la couche MAC, le processus d'association ne fait que récupérer cette liste. Une fois le routeur choisi et la demande d'association envoyée, le nœud passe dans l'état $Wait\_for\_reply$ en armant un timer d'attente. Une fois la réponse d'association reçu avant l'expiration du timer, le nœud finalise l'association et passe à l'état associé, *associated*. Si le timer expire avant la réception de la réponse à l'association, le processus repasse à l'état de sélection du père, $father\_choice$. Notons que cette procédure ne prend pas en compte le cas où le nœud n'arrive pas à trouver un père. Dans un tel cas, le nœud ne sortira jamais de cette procédure et n'est donc pas capable d'émettre et de recevoir des données.

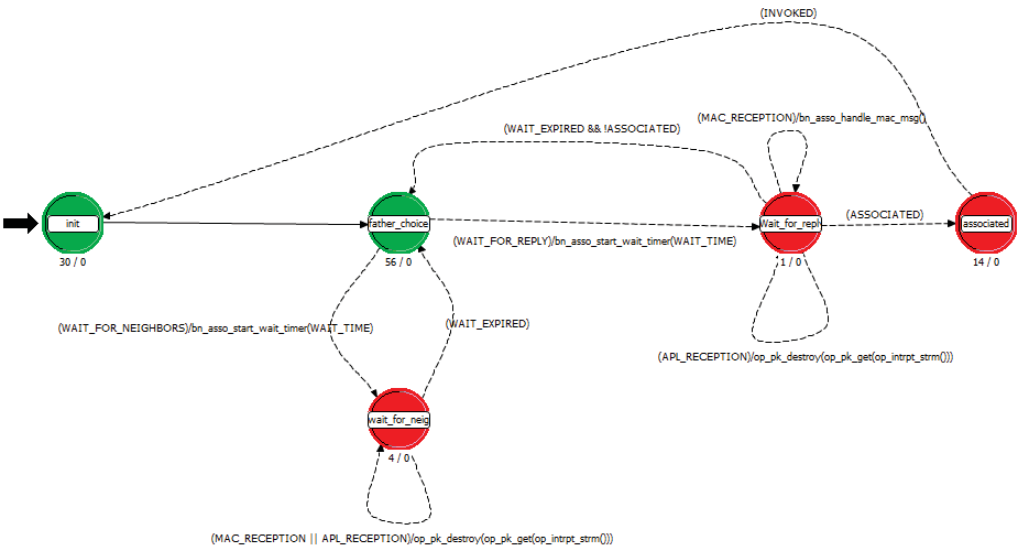


FIGURE B.7 – Modèle de processus de la couche réseau.

B.2.4 Couche application

Cette couche comporte deux modules ; *gen* permettant de générer des paquets selon la loi de probabilité et les paramètres associés choisies et *sink* permettant de recevoir et de détruire un paquet reçu. Les modèles de processus correspondants sont ceux fournis par OPNET.

B.3 S-CoSenS

Le modèle d'un nœud S-CoSenS est identique à celui de CoSenS. Seuls les modèles des processus MAC et réseaux sont différents. Cependant, le modèle de la couche réseau est identique à celui de CoSenS à part que l'association se fait dans la couche MAC. On présente par la suite le modèle de la couche MAC pour les routeurs car celui des nœuds simples n'est qu'une version allégée et un peu modifiée du premier (sans la structure de la supertrame et ne se réveille qu'à la réception d'un paquet de la couche application).

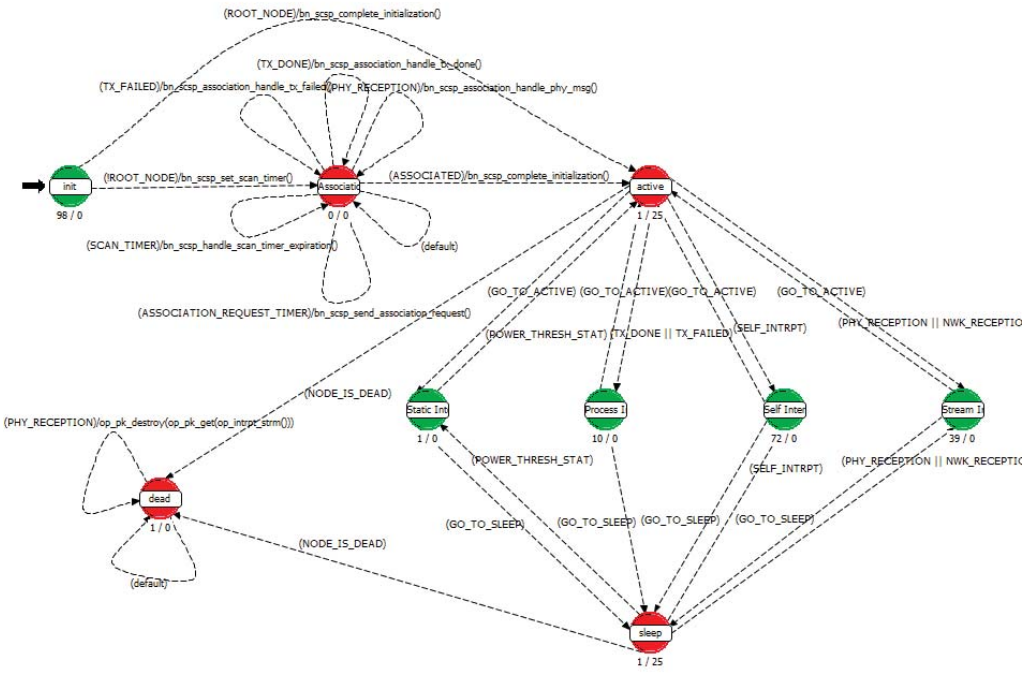


FIGURE B.8 – Modèle de processus de la couche réseau.

Le modèle de processus de la couche MAC est illustré par la figure B.8. Il est composé de quatre états importants, *association*, *active*, *sleep* et *dead* ainsi que cinq états intermédiaires, *init*, *Static Interrupt*, *Process Interrupt*, *Self Interrupt* et *Stream Interrupt*. Les trois derniers états intermédiaires permettent le passage de l'état réveillé, *active*, à l'état d'endormement, *sleep*. *Static Interrupt* gère les événements correspondant à la réception de la puissance de réception du dernier paquet

reçu par la couche physique. *Process Interrupt* gère les événements générés par le processus fils CSMA/CA (succès ou échec de transmission). *Self Interrupt* gère les événements crées par le nœud lui même. Principalement, cela correspond aux événements d'expiration des timers. *Stream Interrupt* gère les événements correspondant à la réception d'un paquet par les couches réseau et physique.

Le fonctionnement complet de S-CoSenS est détaillé dans le chapitre 5. Dans ce qui suit, nous présentons la manière avec laquelle il est implémenté à travers un exemple type ; des paquets sont reçus mais aucun pendant la SP et toutes les transmissions se font avec succès. Après l'initialisation des paramètres du nœud dans l'état *init* et son association dans l'état *association*, il passe à l'état *actif*. Dans cet état, il commence par envoyer un beacon en utilisant CSMA/CA (tout autre paquet reçu est ou bien traité ou bien mis dans la file d'attente). A la réception du résultat de l'opération de transmission du beacon par CSMA/CA, on passe à l'état *Process Interrupt*, qui, en cas de succès, arme le timer d'endorment périodique ainsi que celui de la SP et décide de passer à l'état d'endormement, *sleep* (en attribuant la valeur *True* à *GO_TO_SLEEP* et *False* à *GO_TO_ACTIVE*). A l'expiration du timer d'endormement périodique, le processus passe à l'état *Self Interrupt* qui détermine la nature de l'événement, en l'occurrence expiration du timer d'endormement périodique, et appelle la fonction correspondante qui dans ce cas vérifie l'état du canal et décide de passer à l'état actif dans le cas où le canal est occupé (réception d'un préambule). Si le canal est libre, le timer d'endormement périodique est armé de nouveau et le processus passe à nouveau à l'état *sleep*. A l'expiration du timer de la SP, le nœud se réveille à nouveau en passant toujours par l'état *Self Interrupt* qui, cette fois arme le timer de WP. A l'expiration du timer de WP, le nœud passe à la TP toujours en passant par l'état *Self Interrupt* qui cette fois appelle la fonction de gestion de la TP et revient à l'état *active*. Le premier appel à la fonction de gestion de la TP récupère le premier paquet de la file d'attente et invoque CSMA/CA. Elle est à nouveau appelée par l'état *Stream Interrupt* à la réception de l'acquittement. Une fois tous les paquets transmis, le beacon est transmis à nouveau et les timer d'endormement périodique et de la SP sont armés. Notons que le processus passe à l'état *Process Interrupt* à chaque fois qu'un paquet provenant des autres couches est reçus, un traitement différent est effectué selon que le processus était réveillé ou dormant. Sachant que l'énergie diminue pendant chaque opération, le processus passe à l'état *dead* lorsque celle-ci devient nulle.

Annexe C

Implémentation de CoSenS sur les capteurs Jennic

Les figures C.1, C.2 et C.3 présentent les diagrammes de flux d'un coordinateur du PAN, d'un routeur et d'un nœud simple correspondant à l'implémentation de CoSenS sur Jennic. Rappelons, que CoSenS a été implémenté au dessus de la couche MAC de IEEE 802.15.4. Le programme développé se base sur la file des évènements, une API permettant de regrouper les interruptions générées par la couche MAC et le matériel et les envoyer à la couche supérieure un par un [Jennic Ltd. 2011b]. Une interruption est un signal émis par un programme ou un matériel pour indiquer qu'un événement vient de survenir, comme par exemple la réception d'un paquet ou l'expiration d'un timer. Les interruptions sont regroupées en trois catégories et une file est attribuée à chaque catégorie ; une pour les interruptions du MLME (MAC sublayer management entity), une pour les interruptions du MCPS (MAC common part sublayer) [IEEE-TG15.4 2006] et une pour les interruptions générées par le matériel. Cependant, cette file ne gère pas toutes les interruptions, certaines d'entre elles, comme l'interruption générée par le timer "tick-timer" et utilisée par les routeurs et le coordinateur, doit être gérée indépendamment. Dans ce cas, il s'agit d'affecter une fonction au timer, dite une "callback_function", qui sera appelée à chaque interruption générée par le "tick_timer".

Après le démarrage des nœuds et l'initialisation des paramètres, le coordinateur du PAN démarre la procédure de création du réseau et les autres celle d'association. La première étape de cette procédure pour le coordinateur du PAN est l'*energie_scan* à la recherche du meilleur canal. Pour les autres c'est la procédure *active_scan* à la recherche de pères. Ensuite, tous les nœuds démarrent la boucle infinie contenant une procédure qui vérifie constamment l'état de la file des événements et prends les actions nécessaires ainsi que la fonction de début de la TP activée lorsque $tick\_timer\_fired = TRUE$ uniquement pour les routeurs et le coordinateur.

L'interruption "tick_timer" peut être générée à n'importe quel moment de l'exécution. La fonction appelée incrémente la variable *NbTicks*, vérifie si elle a atteint *NbTicksMax*. Si tel est le cas, elle activera "tick_timer_fired" et arrêtera le "tick_timer". "tick_timer_fired" annonce la fin de la WP et permet de démarrer la procédure d'envoie du premier paquet (début de la TP). Le reste des paquets étant envoyés après la réception de l'acquittement. "NbTicksMax" est estimée avant le début de chaque WP en comptabilisant les messages reçus et en appelant la fonction d'estimation. La durée du "tick_timer" est de 1 ms. Celle de "sensor_timer"

est fixée selon le temps d'inter-arrivée souhaité.

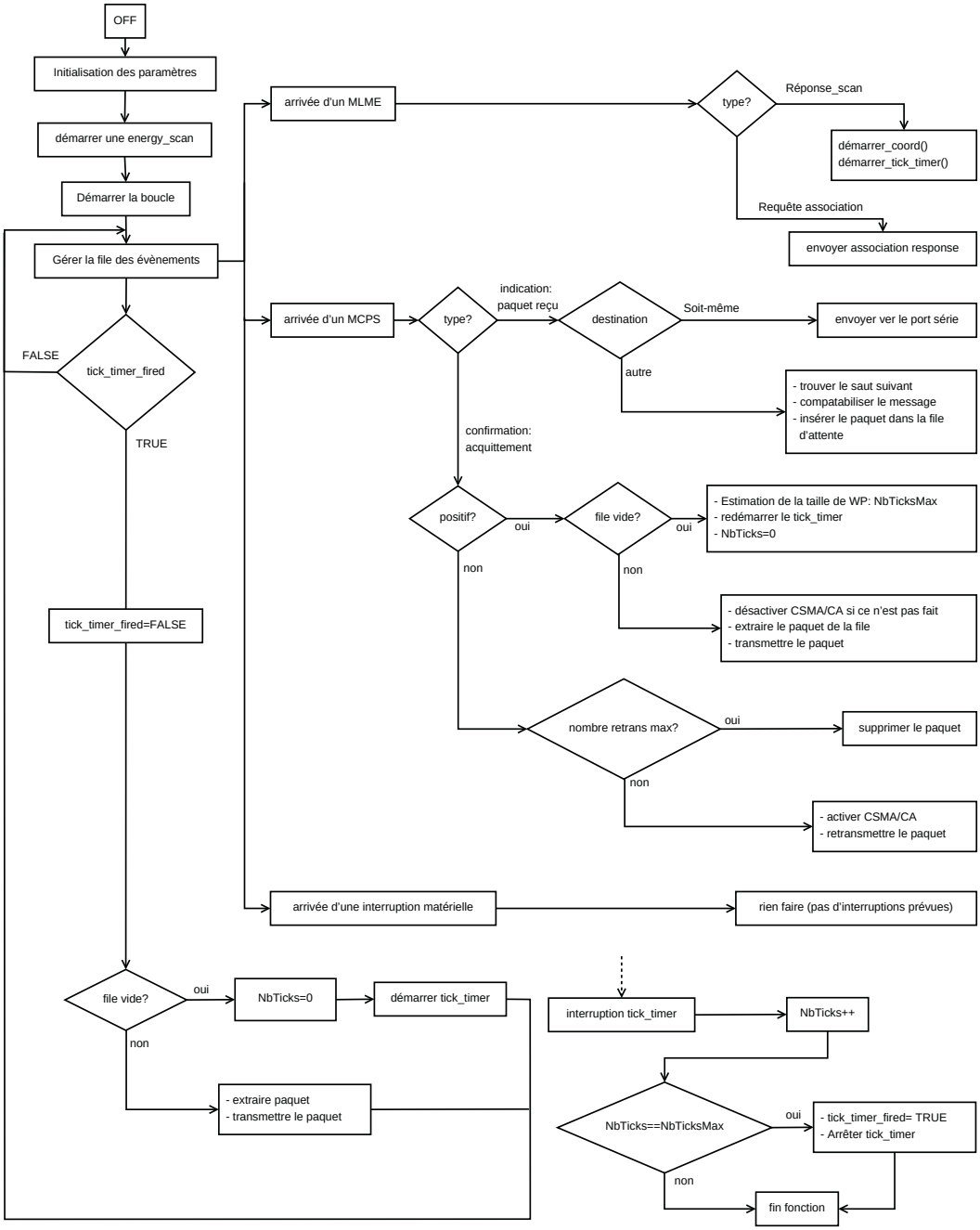


FIGURE C.1 – diagramme de flux du coordinateur du PAN.

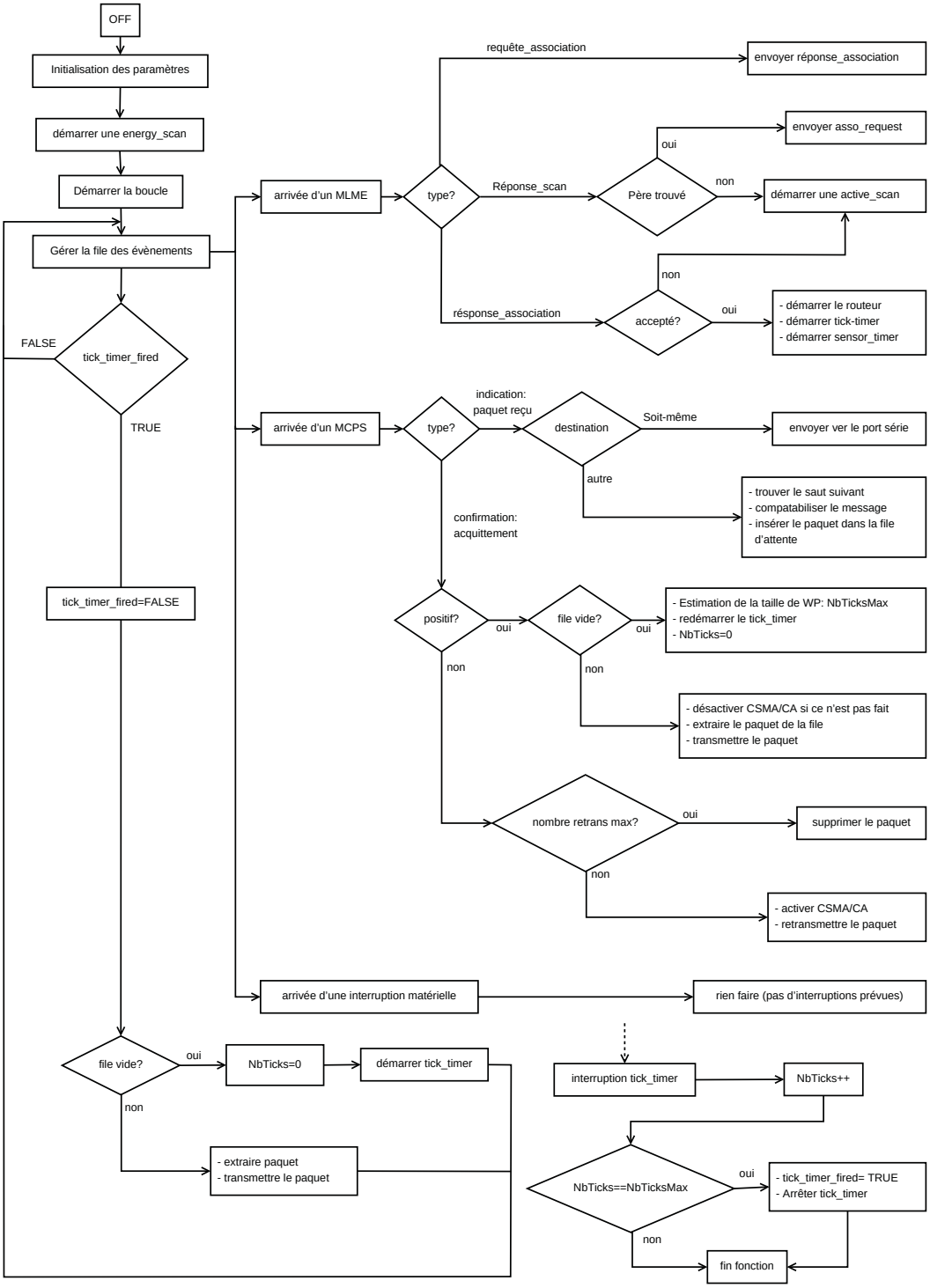


FIGURE C.2 – diagramme de flux du routeur.

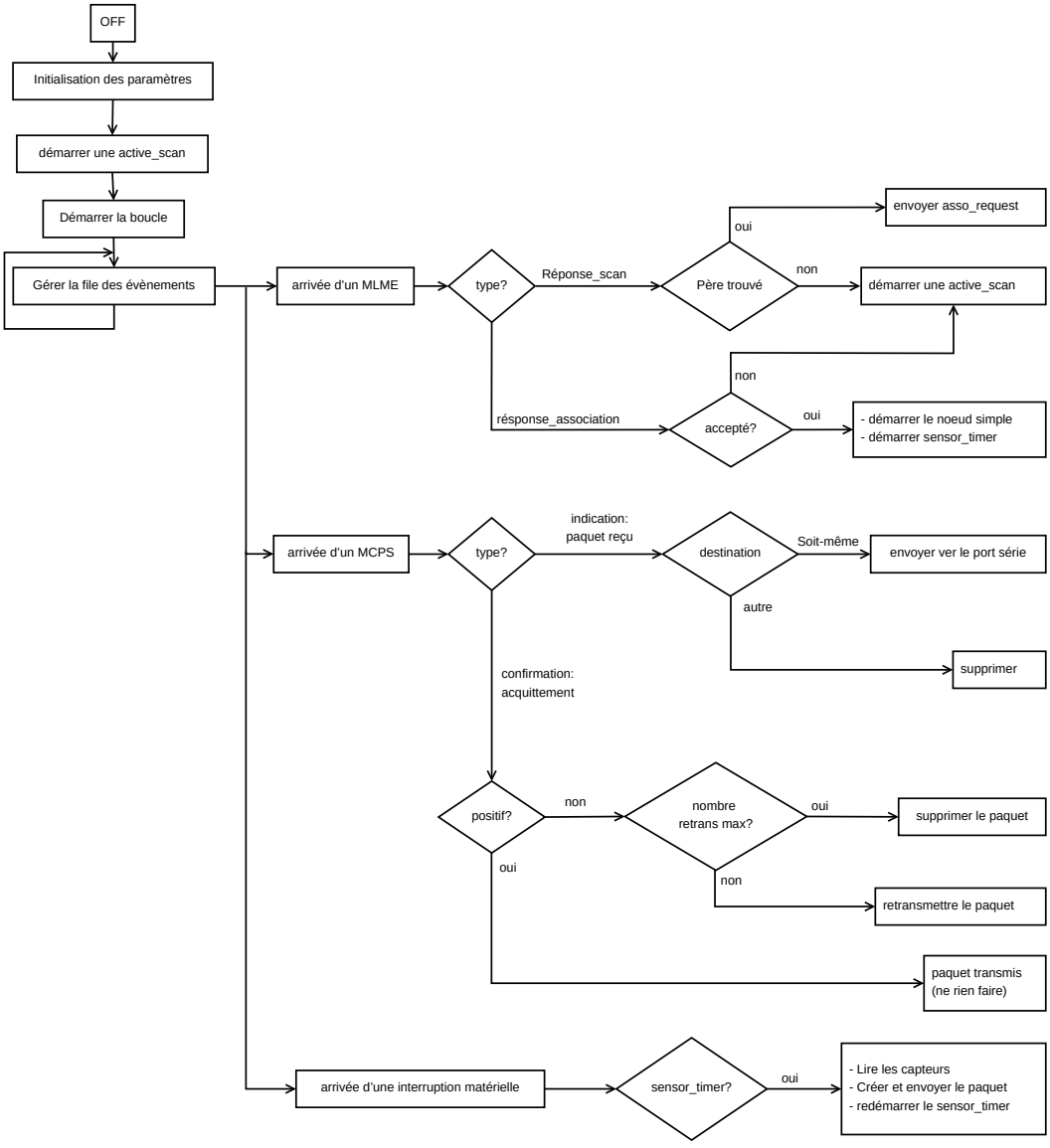


FIGURE C.3 – diagramme de flux du nœud simple.

Table des figures

Liste des tableaux

Liste des publications et des soumissions

Publications

- B. Nefzi, Y.-Q. Song. "QoS for Wireless Sensor Networks using CoSenS : Enabling Service Differentiation in the MAC Sub-Layer". Journal accepté à "SI :Advances in AdHoc Nets(II)". Version étendu du papier "QoS for Wireless Sensor Networks : Tackling Service Differentiation in MAC sub-layer".
- B. Nefzi, Y.-Q. Song. "CoSenS : a Collecting and Sending Burst Scheme for Performance Improvement of IEEE 802.15.4".35th Annual IEEE Conference on Local Computer Networks and Workshops, LCN 2010.
- B. Nefzi, Y.-Q. Song. "QoS for Wireless Sensor Networks : Tackling Service Differentiation in MAC sub-layer", AdHocNets 2010. Second International Conference on Ad Hoc Networks. Sélectionné pour une soumission au journal "SI :Advances in AdHoc Nets(II)".
- B. Nefzi, Y.-Q. Song , "Performance improvement of CSMA/CA for data gathering scenarios in wireless sensor networks," Communications and Networking (ComNet), 2010 Second International Conference on , vol., no., pp.1-7, 4-7 Nov. 2010.
- Nefzi, B. ; Cruz-Sanchez, H. ; Ye-Qiong Song ; , "SCSP : An energy efficient network-MAC cross-layer design for wireless sensor networks," Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on , vol., no., pp.1061-1068, 20-23 Oct. 2009.
- B. Nefzi, Y.-Q. Song. N-MAC : a Network MAC cross layer design for supporting differentiated services and simpler management mechanisms in Wireless Sensor Networks, in 2nd Junior Researcher Workshop on Real-Time Computing (JRWRTC 2008) (2008).
- B. Nefzi, Y.-Q. Song. Performance Analysis and improvement of ZigBee routing protocol, in : 7th IFAC International Conference on Fieldbuses & Networks in Industrial & Embedded Systems - FeT'2007, Toulouse France, 2007.
- A. Koubaa, M. Alves, B. Nefzi, and Y. Q. Song, "Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Time-Critical Events in Wireless Sensor Networks," Workshop on Real Time Networks RTN'06, Jul. 2006.

Bibliographie

[Ahn 2006] Gahng-Seop Ahn, Se Gi Hong, Emiliano Miluzzo, Andrew T. Campbell et Francesca Cuomo. *Funneling-MAC : a localized, sink-oriented MAC for boosting fidelity in sensor networks.* In SenSys '06 : Proceedings of the 4th international conference on Embedded networked sensor systems, pages 293–306, New York, NY, USA, 2006. ACM. (Cité en pages 17 et 19.)

[Akyildiz 2010] Ian Fuat. Akyildiz et Mehmet Can Vuran. Wireless Sensor Networks. John Wiley & Sons Ltd., 2010. (Cité en pages 1 et 2.)

[Bachir 2010] A. Bachir, M. Dohler, T. Watteyne et K.K. Leung. *MAC Essentials for Wireless Sensor Networks.* Communications Surveys Tutorials, IEEE, vol. 12, no. 2, pages 222 –248, quarter 2010. (Cité en page 26.)

[Bertsekas 1992] Dimitri Bertsekas et Robert Gallager. Data networks (2nd ed.). Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992. (Non cité.) :1992 :1992 :1992 :1992

[Bose 2001] Prosenjit Bose, pat Morin, Ivan Stojmenovic et Jorge Urrutia. *Routing with Guaranteed Delivery in ad hoc Wireless Networks.* In Wireless Networks, pages 48–55, 2001. (Cité en page 34.)

[Buettner 2006] Michael Buettner, Gary V. Yee, Eric Anderson et Richard Han. *X-MAC : a short preamble MAC protocol for duty-cycled wireless sensor networks.* In SenSys '06 : Proceedings of the 4th international conference on Embedded networked sensor systems, pages 307–320, New York, NY, USA, 2006. ACM. (Cité en pages 14 et 123.)

[Caccamo 2002] M. Caccamo, L.Y. Zhang, Lui Sha et G. Buttazzo. *An implicit prioritized access protocol for wireless sensor networks.* In Real-Time Systems Symposium, 2002. RTSS 2002. 23rd IEEE, pages 39 – 48, 2002. (Cité en page 19.)

[Chatterjea 2004] S. Chatterjea, L.F.W. van Hoesel et P.J.M. Havinga. *AI-LMAC : an adaptive, information-centric and lightweight MAC protocol for wireless sensor networks.* In Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. Proceedings of the 2004, pages 381 – 388, 2004. (Cité en page 16.)

[Chipara 2006] O. Chipara, Z. He, Guoling Xing, Qin Chen, Xiaorui Wang, Chenyang Lu, J. Stankovic et T. Abdelzaher. *Real-time Power-Aware Routing in Sensor Networks.* In Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on, pages 83 –92, 2006. (Cité en page 38.)

[Dunkels 2011] Adam Dunkels. *Contiki Operating System, http://www.sics.se/contiki/*, Avril 2011. (Cité en page 77.)

[El-Hoiydi 2004] Amre El-Hoiydi et Jean-Dominique Decotignie. *WiseMAC : An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks.*

In Sotiris Nikoletseas et José D. P. Rolim, editeurs, Algorithmic Aspects of Wireless Sensor Networks, volume 3121 of *Lecture Notes in Computer Science*, pages 18–31. Springer Berlin / Heidelberg, 2004. 10.1007/978-3-540-27820-7_4. (Cité en page 13.)

[Ergen 2006] Sinem Coleri Ergen et Pravin Varaiya. *PEDAMACS : Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks*. IEEE Transactions on Mobile Computing, vol. 5, pages 920–930, 2006. (Cité en page 15.)

[Felemban 2006] Emad Felemban, Chang-Gun Lee et Eylem Ekici. *MMSPEED : Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks*. IEEE Transactions on Mobile Computing, vol. 5, pages 738–754, 2006. (Cité en page 37.)

[Ferguson 1998] Paul Ferguson et Geoff Huston. Quality of service : delivering qos on the internet and in corporate networks. John Wiley & Sons, Inc., New York, NY, USA, 1998. (Cité en page 5.)

[Finn 1987] G. G. Finn. *Routing and Addressing Problems in Large Metropolitan-Scale Internetworks*. Rapport technique ISI/RR-87-180, Information Sciences Institute, Mars 1987. (Cité en page 33.)

[Fireflies-rtls 2010] Fireflies-rtls. *http://www.fireflies-rtls.com/*, Mai 2010. (Cité en page 2.)

[Group 1996] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick et V. Jacobson. *RTP : A Transport Protocol for Real-Time Applications*, RFC 1889, 1996. (Cité en page 17.)

[Halkes 2007] G. Halkes et K. Langendoen. *Crankshaft : An Energy-Efficient MAC-Protocol for Dense Wireless Sensor Networks*. In Koen Langendoen et Thiemo Voigt, editeurs, Wireless Sensor Networks, volume 4373 of *Lecture Notes in Computer Science*, pages 228–244. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-69830-2_15. (Cité en pages 17, 18 et 123.)

[HartComm 2007] HartComm. *WirelessHART*, 2007. `http://www.hartcomm.org/index.html`. (Cité en pages 24, 25 et 123.)

[He 2003] Tian He, John A Stankovic, Chenyang Lu et Tarek Abdelzaher. *SPEED : A Stateless Protocol for Real-Time Communication in Sensor Networks*. Distributed Computing Systems, International Conference on, vol. 0, page 46, 2003. (Cité en page 37.)

[He 2006] Tian He, Sudha Krishnamurthy, Liqian Luo, Ting Yan, Lin Gu, Radu Stoleru, Gang Zhou, Qing Cao, Pascal Vicaire, John A. Stankovic, Tarek F. Abdelzaher, Jonathan Hui et Bruce Krogh. *VigilNet : An integrated sensor network system for energy-efficient surveillance*. ACM Trans. Sen. Netw., vol. 2, pages 1–38, February 2006. (Cité en page 2.)

[Hedetniemi 1988] Sandra M. Hedetniemi, Stephen T. Hedetniemi et Arthur L. Liestman. *A survey of gossiping and broadcasting in communication networks*. Networks, vol. 18, no. 4, pages 319–349, 1988. (Cité en page 28.)

[Heinzelman 2000] W.R. Heinzelman, A. Chandrakasan et H. Balakrishnan. *Energy-efficient communication protocol for wireless microsensor networks*. In System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on, page 10 pp. vol.2, 2000. (Cité en page 31.)

[Hoesel 2004] L. F. W. Van Hoesel et P. J. M. Havinga. *A lightweight medium access protocol (LMAC) for wireless sensor networks : Reducing Preamble Transmissions and Transceiver State Switches*. First International Conference on Networked Sensing Systems, 2004. (Cité en page 15.)

[IEEE-TG15.4 2006] IEEE-TG15.4. *IEEE-TG15.4 (2006). Part 15.4 : Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for low-rate Wireless Personal Area Networks (LR-WPANs)*, September 2006. (Cité en pages 22 et 119.)

[IEEE 2007] IEEE. *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements - Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999), pages C1 –1184, December 2007. (Cité en page 11.)

[IETF 2010] IETF. *RPL : IPv6 Routing Protocol for Low power and Lossy Networks, draft-ietf-roll-rpl-17*, 2010. (Cité en page 43.)

[IETF 2011] IETF. *Routing Over Low power and Lossy networks (roll), http://datatracker.ietf.org/wg/roll/charter/*, June 2011. (Cité en page 27.)

[inf 2010] *Informatique Située - Plate-forme, "Appartement intelligent pour l'assistance à la personne". http://infositu.loria.fr/*, Mai 2010. (Cité en page 2.)

[Intanagonwiwat 2000] Chalermek Intanagonwiwat, Ramesh Govindan et Deborah Estrin. *Directed diffusion : a scalable and robust communication paradigm for sensor networks*. In Proceedings of the 6th annual international conference on Mobile computing and networking, MobiCom '00, pages 56–67, New York, NY, USA, 2000. ACM. (Cité en page 30.)

[Jacobson 1988] V. Jacobson. *Congestion avoidance and control*. SIGCOMM Comput. Commun. Rev., vol. 18, pages 314–329, August 1988. (Cité en page 6.)

[Jennic Ltd. 2011a] Jennic Ltd. *802.15.4 Stack API Reference Manual Revision 1.8 30 September 2008, http://www.jennic.com/files/support_files/JN-RM-2002-802.15.4-Stack-API-1v8.pdf*, Avril 2011. (Cité en page 77.)

[Jennic Ltd. 2011b] Jennic Ltd. *Application Queue API Reference Manual Revision 1.0 21 September 2006, http://www.jennic.com/files/support_files/JN-RM-2025-App-Queue-API-1v0.pdf*, Mai 2011. (Cité en page 119.)

[Jennic Ltd. 2011c] Jennic Ltd. *Jennic website, http://www.jennic.com/*, Avril 2011. (Cité en page 49.)

[Jennic Ltd. 2011d] Jennic Ltd. *JN5148 datasheet v1.6 29th Nov '10, http://www.jennic.com/files/support_files/JN-DS-JN5148-1v6.pdf*, Avril 2011. (Cité en page 73.)

[Johnson 2007] David B. Johnson, David A. Maltz et Yih C. Hu. *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR).* Rapport technique, IETF MANET Working Group, Février 2007. (Cité en page 45.)

[Karl 2004] Holger Karl. *Quality of service in wireless sensor networks : mechanisms for a new concept ?, http://www.cs.uni-paderborn.de/fileadmin/Informatik/AG-Karl/Publications/Talks/karl.pdf*, April 2004. (Cité en page 91.)

[Ko 2010] JeongGil Ko, Chenyang Lu, M.B. Srivastava, J.A. Stankovic, A. Terzis et M. Welsh. *Wireless Sensor Networks for Healthcare.* Proceedings of the IEEE, vol. 98, no. 11, pages 1947 –1960, nov. 2010. (Cité en page 71.)

[Koubaa 2006] Anis Koubaa, Mário Alves, Bilel Nefzi et Ye-Qiong Song. *Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Time-Critical Events in Wireless Sensor Networks.* In Proceedings of the Workshop of Real-Time Networks (RTN 2006), Satellite Workshop to ECRTS 2006, Dresden, Germany, 07 2006. Jean-Dominique Decotignie. (Non cité.) :2006 :2006 :2006 :2006

[Kranakis 1999] Evangelos Kranakis, School Of Computer Science, Harvinder Singh et Jorge Urrutia. *Compass Routing on Geometric Networks.* In in Proc. 11 th Canadian Conference on Computational Geometry, pages 51–54, 1999. (Cité en pages 33 et 34.)

[Kulik 2002] Joanna Kulik, Wendi Heinzelman et Hari Balakrishnan. *Negotiation-based protocols for disseminating information in wireless sensor networks.* Wirel. Netw., vol. 8, pages 169–185, March 2002. (Cité en page 28.)

[Li 2005] Yuan Li, Wei Ye et J. Heidemann. *Energy and latency control in low duty cycle MAC protocols.* In Wireless Communications and Networking Conference, 2005 IEEE, volume 2, pages 676 – 682 Vol. 2, 2005. (Cité en page 18.)

[Li 2007] Jian Li. *Garantir la qualité de service temps réel selon l'approche (m,k)-firm.* These, Institut National Polytechnique de Lorraine - INPL, Février 2007. (Non cité.) :2007 :2007 :2007 :2007

[Li 2009] Yanjun Li, Chung Shue Chen, Ye-Qiong Song, Zhi Wang et Youxian Sun. *Enhancing Real-Time Delivery in Wireless Sensor Networks With Two-Hop Information.* Industrial Informatics, IEEE Transactions on, vol. 5, no. 2, pages 113 –122, may 2009. (Cité en page 37.)

[Lindsey 2002] S. Lindsey et C.S. Raghavendra. *PEGASIS : Power-efficient gathering in sensor information systems.* In Aerospace Conference Proceedings, 2002. IEEE, volume 3, pages 3–1125 – 3–1130 vol.3, 2002. (Cité en page 32.)

[Liu 1973] C. L. Liu et James W. Layland. *Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment.* J. ACM, vol. 20, pages 46–61, January 1973. (Cité en pages 5, 19 et 80.)

[Liu 2005] Yang Liu, I. Elhanany et Hairong Qi. *An energy-efficient QoS-aware media access control protocol for wireless sensor networks.* In Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on, pages 3 pp. –191, nov. 2005. (Cité en page 19.)

[Liu 2006] Zhenzhen Liu et I. Elhanany. *RL-MAC : A QoS-Aware Reinforcement Learning based MAC Protocol for Wireless Sensor Networks.* In Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on, pages 768 –773, 0-0 2006. (Cité en page 20.)

[Mainwaring 2002] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk et John Anderson. *Wireless sensor networks for habitat monitoring.* In Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM. (Cité en page 2.)

[Micaz 2011] Micaz. *Micaz datasheet,* `http://www.memsic.com/`, 2011. (Cité en page 99.)

[Nefzi 2008] Bilel Nefzi et Ye-Qiong Song. *N-MAC : a Network MAC cross layer design for supporting differentiated services and simpler management mechanisms in Wireless Sensor Networks.* In 2nd Junior Researcher Workshop on Real-Time Computing - JRWRTC 2008 Proceedings of the 2nd Junior Researcher Workshop on Real-Time Computing, pages 17–21, Rennes France, 2008. (Cité en page 55.)

[OPNET Technologies] OPNET Technologies. *OPNET Modeler v15.0.* `http://www.opnet.com`. (Cité en pages 60, 112 et 125.)

[Parekh 1993] Abhay K. Parekh et Robert G. Gallager. *A generalized processor sharing approach to flow control in integrated services networks : the single-node case.* IEEE/ACM Trans. Netw., vol. 1, pages 344–357, June 1993. (Non cité.) :1993 :1993 :1993 :1993

[Perkins 2003] C. Perkins, E. Royer et S. Das. *RFC 3561 Ad hoc On-Demand Distance Vector (AODV) Routing.* Rapport technique, 2003. (Cité en page 41.)

[Polastre 2004] Joseph Polastre, Jason Hill et David Culler. *Versatile low power media access for wireless sensor networks.* In SenSys '04 : Proceedings of the 2nd international conference on Embedded networked sensor systems, pages 95–107, New York, NY, USA, 2004. ACM. (Cité en pages 12 et 92.)

[Rhee 2006] Injong Rhee, Ajit Warrier, Jeongki Min et Lisong Xu. *DRAND : distributed randomized TDMA scheduling for wireless ad-hoc networks.* In MobiHoc '06 : Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, pages 190–201, New York, NY, USA, 2006. ACM. (Cité en page 17.)

[Rhee 2008] Injong Rhee, A. Warrier, M. Aia, Jeongki Min et M.L. Sichitiu. *Z-MAC : A Hybrid MAC for Wireless Sensor Networks.* Networking, IEEE/ACM Transactions on, vol. 16, no. 3, pages 511 –524, jun. 2008. (Cité en pages 16 et 17.)

[Slimane 2010] J.B. Slimane, Ye-Qiong Song, Anis Koubâa et M. Frikha. *Energy Evaluation of PMCMTP for Large-Scale Wireless Sensor Networks.* In Systems and Networks Communications (ICSNC), 2010 Fifth International Conference on, pages 21 –30, aug. 2010. (Cité en page 26.)

[Sohrabi 2000] K. Sohrabi, J. Gao, V. Ailawadhi et G.J. Pottie. *Protocols for self-organization of a wireless sensor network.* Personal Communications, IEEE, vol. 7, no. 5, pages 16 –27, Octobre 2000. (Cité en page 36.)

[Sutton 2005] Richard S. Sutton et Andrew G. Barto. *Reinforcement Learning : An Introduction.* `http://webdocs.cs.ualberta.ca/~sutton/book/ebook/the-book.html`, January 2005. (Cité en page 20.)

[van Dam 2003] Tijs van Dam et Koen Langendoen. *An adaptive energy-efficient MAC protocol for wireless sensor networks.* In SenSys '03 : Proceedings of the 1st international conference on Embedded networked sensor systems, pages 171–180, New York, NY, USA, 2003. ACM. (Cité en pages 11, 12 et 123.)

[YANG 2011] Fei YANG. *RELIABLE AND TIME-CONSTRAINED COMMUNICATION IN WIRELESS SENSOR NETWORKS.* PhD thesis, INSA de Lyon - INRIA Rhône-Alpes, January 2011. (Cité en page 45.)

[Ye 2001] Fan Ye, A. Chen, Songwu Lu et Lixia Zhang. *A scalable solution to minimum cost forwarding in large sensor networks.* In Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on, pages 304 –309, 2001. (Cité en page 36.)

[Ye 2002] Wei Ye, J. Heidemann et D. Estrin. *An energy-efficient MAC protocol for wireless sensor networks.* volume 3, pages 1567 – 1576 vol.3, 2002. (Cité en pages 10, 11, 92 et 123.)

[Yigitel 2011] M. Aykut Yigitel, Ozlem Durmaz Incel et Cem Ersoy. *QoS-aware MAC protocols for wireless sensor networks : A survey.* Computer Networks, vol. In Press, Corrected Proof, pages –, 2011. (Cité en page 27.)

Nancy-Université
INPL

AUTORISATION DE SOUTENANCE DE THESE
DU DOCTORAT DE L'INSTITUT NATIONAL
POLYTECHNIQUE DE LORRAINE

o0o

VU LES RAPPORTS ETABLIS PAR :
Monsieur André-Luc BEYLOT, Professeur, IRIT, ENSEEIHT, Toulouse

Madame Pascale MINET, Chargée de Recherche, INRIA Rocquencourt, Le Chesnay Cedex

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

Monsieur NEFZI Bilel

à soutenir devant un jury de l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE,
une thèse intitulée :

"Mécanismes auto-adaptatifs pour la gestion de la Qualité de Service dans les réseaux de capteurs sans fil. "

en vue de l'obtention du titre de :

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE

Spécialité : **« Informatique »**

Fait à Vandoeuvre, le 30 août 2011
Le Président de l'I.N.P.L.,
F. LAURENT

INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE – NANCY –

Institut National Polytechnique de Lorraine
inpl@inpl-nancy.fr | www.inpl-nancy.fr

2 av. de la Forêt de Haye | BP 3
54501 Vandoeuvre-lès-Nancy Cedex
FRANCE
Tél. : +33(0)3 83 59 59 59
Fax : +33(0)3 83 59 59 55

Résumé : Mécanismes auto-adaptatifs pour la gestion de la Qualité de Service dans les réseaux de capteurs sans fil

La plupart des réseaux de capteurs sans fil d'aujourd'hui fonctionne sur le protocole CSMA/CA. Fournir la qualité de service (QdS) dans un tel réseau est un problème difficile compte tenu de la dynamique du réseau et des contraintes en termes de ressources (énergie et mémoire). Dans cette thèse, sans changer le socle commun du CSMA/CA, nous avons proposé des mécanismes auto-adaptatifs qui permettent de gérer la QdS " best-effort " pour des applications nécessitant de la différenciation de services. Trois mécanismes sont proposés : CoSenS pour "Collecting then Sending burst Scheme", P-CoSenS qui ajoute la gestion de priorités à CoSenS, et S-CoSenS qui ajoute la dimension énergie à CoSenS. La dynamique du réseau est prise en compte grâce à l'auto-adaptation de périodes de collecte et de transmission en rafale. Il est à souligner que le mécanisme CoSenS permet non seulement d'améliorer les performances de CSMA/CA mais aussi de surmonter la difficulté d'ordonnancer les trafics entrant dans un nœud (routeur) car chaque paquet entrant est immédiatement retransmis vers la sortie. En effet, grâce à la période de collecte, les paquets entrants sont mis en file d'attente, rendant ainsi possible d'ordonnancer différemment les paquets selon leur priorité (P-CoSenS). Enfin, le compromis énergie/performance est pris en compte dans S-CoSenS. Selon l'état de l'environnement surveillé, le réseau peut se trouver dans une période où circule un trafic non urgent et souvent faible pendant laquelle il est judicieux de minimiser la consommation d'énergie et une période de trafic important pendant laquelle le réseau doit transporter des données urgentes pour suivre une situation alarmante de plus près. Comme CoSenS, S-CoSenS permet de s'auto-adapter dynamiquement en fonction de ces situations. L'ensemble de nos propositions est validé par simulations et CoSenS est implémenté sur une plateforme de réseau de capteurs.

Mots clés : Réseau de capteur sans fil, QdS, auto-adaptatif, énergie, file d'attente, ordonnancement, optimisation, MAC, routage en arbre.

Abstract : Auto-adaptive mechanisms for Quality of Service management in wireless sensor networks

Nowadays, most of wireless sensor networks use CSMA/CA protocol. Providing quality of service (QoS) support in such networks is a difficult problem because of the network dynamics and the high constraints in terms of resources like energy and memory. In this thesis and without changing the basic access protocol, CSMA/CA, we developed auto-adaptive mechanisms for best-effort QoS targeted to applications requiring differentiation services. Three mechanisms are proposed ; CoSenS for "Collecting then Sending burst Scheme" which enhances the performances of CSMA/CA, P-CoSenS for "Priority CoSenS" which adds priority management to CoSenS and S-CoSenS for "S-CoSenS" which tackles energy efficiency using CoSenS. We emphasize that CoSenS mechanism not only enhances the performance of CSMA/CA in terms of throughput, end-to-end delay and successful transmission rates but enables also the implementation of scheduling policies ; since the router collects and queues paquets before retransmitting them, it has a complete knowledge about them and can then schedule them efficiently (P-CoSenS). S-CoSenS tackles the energy/throughput tradeoff. In fact, the state of a network varies according to the monitored environment. The network may be at some times lightly loaded with non urgent traffic and at other times highly loaded with urgent traffic generated by nodes in order to monitor a happening phenomenon more closely and hence take the best decision. In the former case, the network must save energy. In the latter case, it must provide bandwidth. S-CoSenS auto-adapts its sleeping and active periods according the incoming traffic. Our propositions are validated by simulations and CoSenS is implemented on a wireless sensor platform.

Keywords : wireless sensor network, QoS, auto-adaptability, energy, queue, scheduling, optimization, MAC, tree routing.