Réseaux de capteurs pour l'assistance aux personnes : conception et développement de mécanismes de fiabilisation

Bastien Mainaud

Ecole Doctorale EDITE

Thèse présentée pour l'obtention du diplôme de
DOCTEUR DE TELECOM & MANAGEMENT SUDPARIS

Doctorat délivré conjointement par
Télécom & Management SudParis *et l'Université Pierre et Marie Curie - Paris 6*

Spécialité :
Informatique - Réseaux

Par
Bastien Mainaud

Réseaux de capteurs pour l'assistance aux personnes : Conception et développement de mécanismes de fiabilisation

Soutenue le 22 juillet 2010 devant le jury composé de :

Pascal Lorentz	Rapporteur, Professeur à l'Université de Haute-Alsace
Hassnaa Moustafa	Rapporteur, Ingénieur de recherche à Orange Labs
Nada Golmie	Examinateur, Manager au NIST -USA-
Guy Pujolle	Examinateur, Professeur à l'Université Pierre et Marie Curie
Yacine Ghamri-Doudane	Examinateur, Maitre de conférence à l'ENSIIE
Vincent Gauthier	Examinateur, Maitre de conférence à Télécom Sudparis
Hossam Afifi	Directeur de thèse, Professeur à Télécom SudParis

Thèse n°
2010TELE0015

Résumé

Les réseaux de capteurs ont créé un domaine de recherche très intéressant avec un champs d'applications très large. L'assistance aux personnes est notamment une des nombreux champs potentiels. Les contraintes de ce type de réseaux sont nombreuses et nécessitent des mécanismes spécifiques. Les problématiques de consommation d'énergie et de sécurité sont particulièrement importantes et ont fait l'objet de travaux spécifiques. La thématique de l'assistance aux personnes impose notamment des contraintes de robustesse et de fiabilité des communications.

Le but de ces travaux de recherche est de définir diverses solutions permettant de répondre à ces problématiques. Ces travaux se décomposent en trois parties.

Dans un premier temps, une plate-forme de communications basée sur les réseaux personnels PAN a été définie. Cette plate-forme a été développée et intégrée dans une station de métro Parisienne. Une modélisation de cette plate-forme ainsi qu'une analyse des observations et des résultats issus de cette intégration nous ont permis d'identifier les faiblesses de l'architecture et des technologies mises en oeuvres.

Dans un deuxième temps, nous avons développé diverses solutions permettant de fiabiliser cette plate-forme. En particulier, nous avons conçu un algorithme d'ordonnancement permettant de réduire la consommation d'énergie dans les réseaux capteurs par l'utilisation d'une analyse sémantique des données. Nous avons ensuite proposé une architecture de sécurité, Tiny 3-TLS, qui permet de sécurisé les communications entre un capteur et une entité située sur un réseau disjoint.

Enfin, nous nous sommes intéressés aux communications entre cette plate-forme et les terminaux mobiles. L'aspect fiabilité a en particulier fait l'objet de travaux spécifiques. C'est pourquoi dans un troisième temps, nous avons proposé une solution de routage définissant une nouvelle métrique. Nous avons ensuite proposé un protocole coopératif permettant un apport de fiabilité dans les communications.

Abstract

Wireless Sensors Networks is a very active research area with a very large scope of possible application. In particular, assistance to person is one of the promising applications.

The constraints in this kind of networks are strong and needs of specifics mechanismes. Energy consumption and security are particularly important and are subject of several works. Assistance to person set some robustness constraints and communications reliability. The goal of theses works is to define several solutions to answer to these problematics.

This work has been split in three differents parts.

First, a communication platform based on Personal Network technics was defined. This platform was design, developp and install in a Paris metro station. We made an analytical model of our platform to perform an analysis of its performances. This analysis and the results from the platform installation itself allow us to clearly define the architecture weaknesses and the technologies drawbacks.

Second, we developped several solutions to bring reliability to the plarform. First, we designed a scheduling algorithm allowing to reduce the bandwidth utilization. This algorithm uses a semantic data analysis to help the data scheduling. Then, we define a solution bringing security to the communications between a sensor node and a remote monitoring terminal. This solution allows to establish a secure end-to-end tunnel between two entities in an heterogenous network.

At last, we were interested to the communications between the platform and the mobile terminal. We focus our work on the reliability aspect of the communications. So, in the last part of this thesis, we suggest a routing protocol using a metric based on the RSSI. Then, we define a cooperative protocol bringing more reliability to the communications.

Remerciements

Je tiens à exprimer tout d'abord mes remerciements aux membres du jury, qui ont accepté d'évaluer mon travail de thèse.

Merci à Madame Hassnaa Moustafa et Monsieur Pascal Lorenz, d'avoir accepté d'être les rapporteurs de ce manuscrit. Leurs remarques et suggestions lors de la lecture de mon rapport m'ont permis d'apporter des améliorations à la qualité de ce dernier.

Merci également à Madame Nada Golmie, monsieur Guy Pujolle et monsieur Yacine Gahmri Doudane pour avoir accepté d'examiner mon mémoire et de faire partie de mon jury de thèse.

Merci à Hossam Afifi, pour avoir accepté d'encadrer cette thèse et dont l'aide précieuse m'a été indispensable sur le plan scientifique. Je tiens également à le remercier pour la confiance et la sympathie qu'il m'a témoignées au cours de cette thèse. Merci également à Vincent Gauthier pour son aide lors de la dernière année de ma thèse.

Mes sincères remerciements à Djamal Zeghlache, responsable du laboratoire RS2M de l'Institut Telecom SudParis, de m'avoir accueilli au sein de son équipe.

Je tiens à remercier l'ensemble de l'équipe et plus particulièrement Isabelle Rebillard pour sa gentillesse et son efficacité lors des difficultés administratives ou logistiques que j'ai rencontrées.

Je tiens enfin à remercier les amis, thésards ou non qui m'ont aidé au cours de cette thèse : Vincent V., Mehdi M., Inès, Mariem, Teck, Sepideh, Chedly, Mohamed Abid, Anahita, Makhlouf, Khaled, Mohamed Boutabia, Aroua, Ahmad.

Finalement j'adresse un grand merci à toute ma famille qui a toujours été présente lorsque j'en ai eu besoin, en particulier à ma femme qui a su me soutenir pendant toutes ces années.

Table des matières

Table des figures

Liste des tableaux

Introduction

Aujourd'hui, permettre au plus grand nombre de personnes d'être autonome est un challenge aussi bien sociologique, éthique que technologique. L'assistance aux personnes est un thème très général qui regroupe différentes parties :

- La télémedecine : elle permet par exemple de surveiller les constantes médicales d'un patient ou d'établir un diagnostic à distance [25].
- L'aide aux déplacements pour des personnes à mobilité réduite : il peut s'agir dans ce cas, d'aider ces personnes à trouver les équipements (*e.g.* ascenseur) leur permettant de se déplacer plus facilement ou de situer des services (distributeur de billets, comptoir d'accueil) qui leur soient facilement accessibles.
- L'aide à la localisation pour des personnes déficient visuelles : elle peut aider à se guider au sein d'un lieu inconnu par exemple [70].

Mais cette assistance pourrait également se faire par le biais d'outils de communications permettant aux personnes de localiser des lieux et des services ou de se situer au sein d'un lieu inconnu.

Dans notre cas, cette assistance a fait l'objet de la conception et du développement d'une plate-forme de communications sans fils intégré au sein d'un environnement clos.

Le développement d'une telle plate-forme comporte diverses problématiques. On pense notamment à la latence ou au délai d'acheminement des données. En effet, celle-ci ayant pour objectif de fournir un moyen de communication entre des personnes, la réactivité de la plate-forme est essentielle. Dans notre étude, cette plate-forme était installée au sein d'une station de métro, un milieu clos soumis à de fortes contraintes et dont la mobilité des personnes entraîne une grande complexité dans l'acheminement des données. En effet, contrairement à une environnement ouvert, ou les transmissions subissent un affaiblissement progressif [62], dans un environnement clos tel qu'une gare souterraine, le déplacement des personnes peuvent amener les transmissions à se dégrader très rapidement ou à disparaître complètement sur une très courte période de temps. Les problématiques de découvertes de services ou de personnes subissent également de fortes contraintes et doivent être très réactives. Il est donc nécessaire de définir une architecture efficace permettant d'assurer une couverture maximale de la station ainsi que des mécanismes permettant de palier aux problématiques de transmissions que l'on vient d'exposer.

Dans le cadre de mes recherches, je me suis concentré sur l'utilisation des réseaux sans fils personnels (WPAN) et des réseaux de capteurs sans fils (WSN) [4] pour assurer cette mission. Dans mon travail, le me suis intéressé à différents concepts de réseaux sans fils et à différentes technologies existantes :

- Les réseaux de capteurs sans fils et la norme IEEE 802.15.4 (Zigbee),
- Les réseaux personnels Bluetooth,
- Les réseaux locaux sans fils Wifi (IEEE 802.11).

Les améliorations technologique récentes ont permis de développer des terminaux de communications petits, économiques et peu consommateurs d'énergie. Ces terminaux sont plus communément dénommés *sensors*, capteurs. Chacun de ces capteurs est capable de réaliser différentes taches, notamment la mesure de variable environnementale (température, pression, luminosité, ...)[91, 7] et de communiquer avec les autres capteurs qui composent son voisinage. Ce voisinage est alors dénommé réseau de capteur et a pour but de distribuer une tache sur plusieurs entités afin de la réaliser.

Un réseau personnel s'entend, dans ce manuscrit, comme étant un réseau dont la taille est moindre qu'un réseau de capteurs et dont l'utilisation n'est pas limitée à la collecte d'informations. Dans un réseau personnel, les données peuvent être amenées à circuler dans les deux sens, c'est à dire d'un terminal (*i.e.* capteur) vers une unité de traitement de l'information et inversement.

Enfin, on observera que d'un point de vue standardisation, la norme IEEE 802.15.4 [44] qui est le standard le plus communément utilisé pour les réseaux de capteurs est lui-même un groupe faisant partie de la norme IEEE 802.15 dont le domaine de compétence vise les réseaux personnels (WPAN).

Ces technologies sont cependant caractérisées par de fortes contraintes. Les réseaux sans fils sont notamment concernés par des problèmes tels que la propagation du signal et la qualité du signal reçu ou la connectivité des noeuds composant le réseau [19]. Dans le cas de réseaux de capteurs, on peut également rajouter des problèmes de robustesse, de consommation des ressources, d'autoconfiguration [87]. De part leur conception et les environnements dans lesquels ils évoluent, les capteurs doivent disposer de solutions permettant d'assurer la robustesse de leur fonctionnement [34, 87, 14]. De même, leur autonomie entraîne des problématiques de consommation d'énergie car ils ne peuvent pas être rechargés. Dans le cas d'un réseau composé de plusieurs milliers de noeuds, ils ne peuvent être initialisés *à la main* et doivent être capables de définir eux même les paramètres leur permettant de fonctionner tel que leur adresse (couche réseau) ou leur noeud source (*sink*).

Nous présenterons alors une architecture spécifique de communications permettant l'assistance aux personnes. Cette architecture devra en particulier répondre aux problématiques de scalabilité et de robustesse.

Dans mon manuscrit, je résous un certain nombre de problèmes dans ce cadre là. Le manuscrit est divisé en 2 parties. La première partie est composée de 3 chapitres.

Dans le premier chapitre de cette partie, je considère une architecture de communications. Dans un souci d'apporter une assistance aux personnes dans un lieu public à grande affluence, j'ai proposé une architecture basée sur la technologie Bluetooth. Cette architecture apporte la personnalisation de services à des personnes ainsi que la communication entre ces personnes. Cette architecture, sous la forme d'une plate-forme, a été déployée et testée au sein d'une station du métro parisien. J'ai ensuite modélisé cette plate-forme afin de valider les choix et de constater les limites du système. A la fois, les tests en grandeur nature et le modèle mathématique confortent la robustesse de ma solution.

Dans le deuxième et troisième chapitre, je considère la terminaison de l'architecture d'assistance aux personnes : le lien entre le terminal de la personne et l'architecture proposée par le biais d'un point d'accès.

Je m'intéresse dans un premier temps aux problématiques de consommation d'énergie et d'utilisation de la bande passante. Je propose un algorithme de décision permettant d'économiser la bande passante et les ressources des capteurs dans un environnement médical utilisant la technologie Zigbee. Cette solution est basée sur un algorithme de compression non destructeur évaluant les données et permettant de définir des règles limitant la transmission des données. Ces données médicales bénéficient notamment d'une pré-analyse avant leur transmission permettant de qualifier leur criticabilité.

Je m'intéresse ensuite aux problèmes de sécurité, de chiffrement et d'authentification. Je résous un problème de sécurité en proposant des mécanismes permettant de sécuriser de bout-en-bout des communications. Cette solution couvre les communications de l'architecture proposé au chapitre 1 jusqu'au capteurs du chapitre 2. Cette solution utilise le principe de délégation de confiance entre différentes entités ne bénéficiant pas toutes des mêmes ressources.

Dans la seconde partie de mon manuscrit, j'étudie le cas de communications entre capteurs sans la présence d'une infrastructure.

Mon but est alors de fiabiliser et d'améliorer le niveau des communications. Je propose donc deux solutions.

Le chapitre 5 présente une solution de communication coopératives adaptées aux réseaux de capteurs. L'intérêt grandissant des communications coopératives dans les réseaux sans fils est qu'il permet d'améliorer significativement les taux de réception des paquets et permet d'économiser de l'énergie.

Les contraintes fortes des réseaux de capteurs obligent a développer des solutions adaptées afin de pouvoir utiliser les principes de la coopération. C'est dans ce sens que nous avons développé une solution de sélection automatique d'un noeud relais

permettant de définir de la façon la plus optimale possible les noeuds participant à la communication coopérative.

Dans le chapitre 6, nous considérons l'utilisation d'une route de secours dans un algorithme de routage pour réseau ad-hoc. Le mécanisme que nous présentons utilise une métrique différente de la métrique classique des réseaux (*i.e.* nombre de sauts). Cette route de secours couplée à cette métrique nous permet de définir une solution de fiabilité des protocoles de routages ad-hoc.

Enfin, nous présenterons nos conclusions ainsi que les perspectives générales envisagées dans le chapitre 7.

Chapitre 1

Etat de l'art

Notre étude porte sur la fiabilité des réseaux de capteurs. Nous avons pour cela développé une plate-forme utilisant la technologie Bluetooth qui est par certains aspects proche de la philosophie des réseaux de capteurs. Cette plate-forme a été enrichi de divers fonctionnalités permettant d'en améliorer la sécurité et les performances réseaux.

Nous avons ensuite étudié l'apport de la technique de communications coopératives dans les réseaux sans fils. Nous avons également analysé les bénéfices de l'utilisation d'une route de secours et d'une métrique plus adapté aux réseaux sans fils lorsque utilisait dans un protocole de routage ad-de type réactif.

Dans ce chapitre, nous présenterons donc un état de l'art des différentes solutions mises en oeuvres ainsi que les différents concepts développés dans la suite de ce document.

1.1 Problématique d'assistance aux personnes et de télémédecine

La télémédecine est une discipline qui a connu de nombreuses évolutions en fonction des avancées technologiques. Actuellement, la technologie émergente des réseaux de capteurs apporte de grands avantages et apparaît comme une solution technologique possible aux problèmes de télémédecine. Cependant, l'environnement particulier qui incombe aux réseaux médicaux et les contraintes des réseaux de capteurs ont obligé la communauté à adapter la technologie à l'environnement. Dans cette section, nous présenterons les différentes problématiques que nous avons étudiées et les travaux existants. Nous nous focaliserons en particulier sur les problématiques de surveillance (*monitoring*), de sécurité et de fiabilité des communications.

1.1.1 Déploiement de réseaux de capteurs pour la télémédecine

Le déploiement de réseaux de capteurs dans un environnement médical permet, en plus de réduire fortement le câblage et donc l'encombrement, d'offrir une plus grande flexibilité aux réseaux. Il n'est plus nécessaire de "tirer" des câbles pour assurer une continuité de service. La surveillance de plusieurs patients peut être réalisée à partir d'une unité centrale de traitement et de capteurs répartis sur chacun des patients.

Néammoins, ce type de déploiement présente certaines contraintes résumées dans [25]. Nous en présentons ici quelques unes.

- **Zone de couverture**
  La zone de couverture est fonction de l'application et de la technologie. Elle peut varier de 1 mètre à 100 mètres. La figure Fig.1.1 présente un exemple de topologie montrant les diverses technologies utilisées dans le cadre d'une surveillance de patient. Il est donc nécessaire, si plusieurs technologies sont mises en place, qu'elles ne se pertubent pas entre elles.
- **Architecture réseaux**
  Deux types de solution peuvent être utilisées : celle basée sur une infrastucture, et celle dépourvue d'infrastructure : ad-hoc. Cette dernière présente l'avantage d'offrir une plus grande flexibilité mais elle est moins facile à administrer. La cohabitation de ces deux types d'architecture est alors nécessaire.
- **Allocation de fréquence**
  Actuellement, les technologies sans fils utilisées pour les réseaux locaux et personnels sont normalisées IEEE 802.11, 802.15.1 et 802.15.4.
  Cependant, certains travaux [36, 86, 92] ont montré que les standards sans fils actuels ne sont pas optimales avec les contraintes inhérentes à ce domaine.

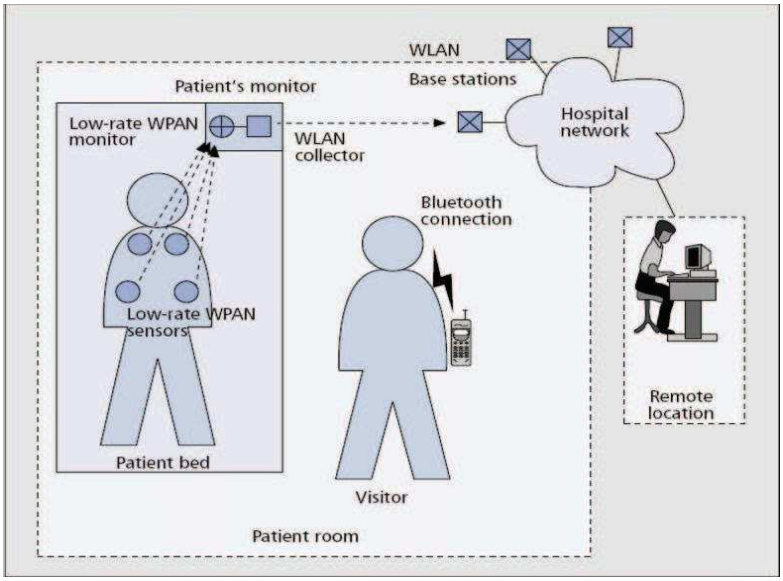

FIG. 1.1 – Exemple de surveillance de patient [25]

Ces études mettent en évidence un problème de passage à l'échelle et une insuffisance de bande passante. Certaines de ces études preconisent par exemple de modifier le temps de backoff du CSMA/CA.
Shnayder *et al.* démontrent dans [86] que le pourcentage de paquets reçus décroit très fortement lorsque l'on augmente le débit et le nombre de noeuds émettant des données. Des résultats identiques sont également observés dans [5], mais Al-Khateeb *et al.* notent que le débit reste relativement constant pour chaque augmentation du nombre de noeuds.
Dans [36], Golmie *et al.* mettent en évidence le lien existant entre le nombre d'émetteurs et le débit au niveau applicatif : à partir de 16 noeuds, le débit est réduit à 5% de la bande passante disponible de la norme 802.15.4.
Afin de réduire la consommation d'énergie et de bande passante, Horspool *et al.* [41] ont proposé une application utilisant la compression des données. Cette solution ne prend néanmoins pas en compte les ressources limitées des capteurs ainsi que le caractère urgent de certaines données qui ne peuvent pas être compressées et doivent être envoyées immédiatement.
Mais, les techniques utilisant cette bande de fréquences sont différentes. Les normes 802.11 et 802.15.4 utilisent la technique d'étalement de spectre DSSS, alors que la norme 802.15.1 utilise la technique de saut de fréquences FHSS. Les normes 802.11 et 802.15.4 utilisant la même bande de fréquences et la même technique de modulation, la cohabitation de ces technologies peut réduire les performances de chacun des réseaux. La norme 802.18 [3] assure la coexistence des

différents technologies de réseaux sans fils. Elle permet de se prémunire des interférences mais celle-ci peuvent tout de même subvenir si l'implémentation n'est pas parfaitement conforme au standard. Petrova *et al.* ont notamment établi dans [77], que les technologies 802.11 et 802.15.4 se perturbent mutuellement. Leur étude s'est principalement concentré sur les interférences générées lorsque l'on utilise les normes 802.11g et 802.11n. Ils établissent notamment que les performances du réseau de capteurs sont liées à la taille du trafic généré par le réseau 802.11. Si le trafic est en deça de 512 kbps, le taux de réception des paquets est proche de 100%. Lorsque le trafic est de 15Mbps, le taux maximum de réception des paquets est alors de 60%. Leurs résultats montrent également l'importance de la taille des paquets à transmettre sur le réseau de capteurs. Les paquets de petite taille (<30 octets) sont à privilégier. Ils établissent aussi que les performances sont liées au seuil de réception *threshold* des capteurs. Si celui-ci peut être dynamiquement défini, les performances du réseau 802.15.4 sont alors améliorées. Yoon *et al.* [110] ont analysé le taux de paquets reçus lorsque la technologie utilisée est normalisée 802.11b. Leur étude tend à montrer qu'une distance de 4 mètres séparant les équipements 802.11 et 802.15.4 permet de limiter les interférences. En deça, leur recommendation porte sur la taille des paquets transmis par le réseau 802.11. Cela permet alors de réduire les interférences sur le réseau de capteurs.
Les choix des canaux, l'utilisation d'une allocation dynamique des fréquences et de la technologie à employer doit donc être réflechi lors d'un deploiement. La réduction du trafic doit aussi être envisagé et des mécanismes le permettant doivent être mise en oeuvre.

Le déploiement de réseaux de capteurs dans un environnement médical permettrait donc d'améliorer la qualité des traitements et de faciliter le travail du personnel médical. De nombreux projets ont étudié l'adaptation et le déploiement de réseaux de capteurs dans des milieux médicaux ou dans des environnement nécéssitant une surveillance médicale. Le projet *CodeBlue* (Wireless Sensor Networks for Medical Care) [22, 34], mené par le département d'ingénierie et de sciences appliquées de l'université d'Harvard, explore les possibles utilisations des réseaux de capteurs pour des applications médicales. Le projet s'intéresse au des scénarios d'urgence(désastre, accident ...), lorsqu'une infrastructure n'est pas déployable rapidement. Dans ce cas, un réseau de capteurs apporterait une certaine flexibilité et permettrait d'accélerer le traitement des patients. Le projet *Proactive Health* d'INTEL [46] étudie de quelles façons l'usage de ces technologies permet d'améliorer le bien-être et la santé des patients chez eux et dans leur quotidien. Les études se focalisent en particulier sur l'aide aux personnes agées et leur rapport à la technologie.

ACTis [72] de l'université d'Alabama et BodyNets [26], de l'Université de Californie à Los Angeles (UCLA), ont implémenté une solution permettant de réaliser une

acquisition de données via un réseau de capteurs (body area network) et d'envoyer les informations collectées vers un pda en utilisant un réseau Wi-Fi.

De nombreux autres laboratoires étudient les problématiques de réseaux de capteurs appliqués à la médecine. La recherche actuelle se concentre sur le développement de réseaux de capteurs portables (wearable) [60, 85, 24] ainsi que sur la surveillance individuelle de patients [61, 101, 8, 111].

1.1.2 Pré-requis pour la télésurveillance des patients

Varshney *et al.*[98] ont défini des pré-requis pour la télésurveillance des patients. Nous présentons dans cette section, une partie des pré-requis sur lesquels nous avons basé notre travail. Ces pré-requis de surveillance sont variés et dépendants du scénario (fréquence de surveillance, volume des données à envoyer ...) et de l'environnement (intérieur, extérieur, mobile ou fixe ...).

1.1.2.1 Télésurveillance et transmission d'informations

On peut citer la pression artérielle, le pouls, la température corporelle, l'electrocardiogramme, l'activité moteur, le taux d'humidité et la température de l'environnement, la détection de fumée etc... On distingue alors deux types de transmissions : la première, qui concerne les signes vitaux et qui doit se faire de façon périodique et la seconde qui concerne l'environnement du patient et qui se fait lorsque cette environnement est modifié.

1.1.2.2 Fiabilité de transmission

La fiabilité des transmissions est primordiale dans le cas d'une télésurveillance. En effet, dans cette situation, la vie d'un patient est potentiellement menacée si une transmission est perdue, retardée ou erronée. Il est donc nécessaire de mettre en place des mécanismes assurant la fiabilité des communications et prenant également en compte les caractéristiques inhérents aux réseaux de capteurs. La fiabilité des communications est aussi fonction des délais de délivrance des données. L'importance des données doit être prise en compte par le protocole de communication et une hiérarchisation des informations doit être définie. Un ralentissement soudain et brutal du rythme cardiaque est probablement une information ayant une priorité supérieure à un relevé de témperature corporelle périodique.

La notion de fiabilité sera développé de façon plus approfondie dans la suite de ce document.

1.1.2.3 Consommation d'énergie

La conservation d'énergie est un facteur prépondérant dans les réseaux de capteurs et encore plus dans ceux dédiés à la surveillance médicale. En effet, il n'est pas envisageable qu'un noeud ne soit plus accessible à cause d'une défaillance d'un noeud intermédiaire ou qu'une mesure ne puisse être effectuée par manque d'énergie. La consommation d'énergie est directement liée à la fiabilité. Un réseau dont les communications seraient *fiables* mais dont la durée de vie serait limitée ne saurait être considéré comme globalement *fiable*. Il faut donc trouver le compromis entre la fiabilisation des transmissions et la durée de vie du réseau.

1.1.2.4 Mobilité

La télésurveillance peut se faire au sein d'un centre de soins mais elle peut également être distribuée dans divers endroits et les patients peuvent ne pas être alités. On peut par exemple penser à la surveillance des constantes de patients se trouvant chez eux et disposant de capteurs mesurant leur environnement et leur constantes vitales. La notion de mobilité des patients et donc intrinséquement des équipements doit donc être étudiée et prise en compte. Il faut également s'intéresser à l'environnement qui peut être intérieur ou extérieur.

1.1.2.5 Confidentialité et respect de la vie privée

Les informations qui circulent sur le réseau ont trait à une personne et contiennent des données sur son état physique. Leur protection doit être une thématique très forte. Les données ne doivent pas pouvoir être consultées et/ou modifiées par une tierce personne.

1.1.3 Sécurité des communications dans les réseaux de capteurs médicaux

La plupart des architectures de réseaux de capteurs sont centralisées. Tous les noeuds envoient leurs informations vers un point central qui traitera les données. Dans le cadre d'un réseau appliqué à la télémédecine, la même type d'architecture est utilisée.

Les premières générations de capteurs n'utilisant pas encore le protocole IP pour communiquer, le déploiement de passerelles entre le réseau de capteurs et le réseau "classique" a été nécessaire. De plus, les noeuds d'un réseau de capteurs ont des ressources limitées, le traitement de l'information est donc réalisé sur une machine ayant des ressources suffisantes (*e.g.* serveur). Cette dernière est alors sur un réseau différent de celui-ci du réseau de capteurs.

Comme présentée dans la précedente section, la notion de sécurité et de confidentialité est primordiale. Il est donc nécessaire de mettre en place une solution de sécurité

FIG. 1.2 – Architecture de Sizzle [37]

prenant en compte la présence d'une passerelle entre les noeuds et le réseau contenant l'unité de traitement de l'information. Il faut également prendre en compte les paramètres matériels. Les noeuds ayant des capacités très limitées, l'utilisation d'algorithmes de sécurité "classique" est proscrite.

De nombreuses solutions de sécurité pour réseaux de capteurs considèrent que les contraintes des noeuds empêchent l'utilisation de cryptographie à clef publique. Mais certains récents travaux ont démontré l'inverse.

Watro *et al.*[103] ont conçu une solution, Tiny PK, basée sur l'utilisation de clefs publiques pour permettre l'authentification et l'échange de clefs entre un réseau de capteurs et une entité se trouvant sur un réseau différent. Cette solution utilise l'algorithme RSA bien qu'il soit considéré "gourmand" en ressources . L'utilisation du chiffrement par courbes elliptiques (*Elliptic Curve Cryptography*) apparaît actuellement comme une alternative au RSA car les clefs sont plus petites pour une sécurité équivalente. Une architecture de sécurité pour capteur, utilisant les courbes elliptiques, a été implémentée par Gupta et al [37] : *Sizzle*. Cette solution permet de pouvoir embarquer un serveur web sécurisé dans un capteur pour réaliser des tâches de surveillance et de contrôle. L'architecure de *Sizzle* (cf Fig.1.2) est composée d'une station de traitement située sur le réseau internet et d'un réseau de capteurs réliés avec internet via une passerelle. Cette dernière dispose d'une interface de type ethernet pour le lien internet et d'une connexion 802.15.4 [44] pour le réseau de capteurs.

Dans cette solution, la passerelle ne réalise aucune opération cryptographique. Toutes les opérations sont réalisées par les capteurs et les messages circulent de manière chiffrée à travers la passerelle. Les données sont donc simplement transmises par cette dernière vers l'unité de traitement. Cela a pour conséquence de surcharger les capteurs alors que la passerelle dispose, dans la plupart des scénarios, de ressources bien supérieures à celles présentes dans les capteurs. De plus, la passerelle ne s'authentifie ni auprès des capteurs ni auprès de l'unité de traitement, une attaque de type *man in the middle* est donc possible.

Il serait donc intéressant de reporter une partie des opérations de chiffrement sur la passerelle et de permettre l'authentification de cette dernière par les diverses entités.

1.2 La norme IEEE 802.15.4

1.2.1 Architecture

Le standard IEEE 802.15.4 définit une couche physique et une couche d'accès au médium (MAC) pour les réseaux personnels à faible débit (*i.e.* Low-Rate Wireless Private Area Networks (LR-WPAN)). Bien qu'il n'est pas été spécifiquement conçu pour les réseaux de capteurs, ce standard définit un pour des équipements à faible capacité (réseau, consommation d'energie, puissance de calcul), ce qui correspond aux caractéristiques des équipements d'un réseau de capteurs. Cette section présente les différentes architectures possibles ainsi que les entités définies par le standard.

1.2.1.1 Entités réseau et Topologies définies par le standard IEEE 802.15.4

1.2.1.1.1 Entités réseau Le standard IEEE 802.15.4 définit deux types d'entités dans un réseau personnel *PAN* :

1.2.1.1.1.1 Full Function Device (FFD) Un FFD est un équipement qui possède trois modes de fonctionnement :

- Un coordinateur de réseau personnel (*PAN Coordinator*) : le contrôleur principal du PAN. Cette entité crée un réseau sur lequel les autres équipements peuvent s'associer.
- Un coordinateur : fournit un service de synchronisation à travers la transmission de balises. Il est associé à un coordinateur de réseau personnel et ne peut pas créer son propre réseau.
- Un équipement simple : tout autre équipement qui ne fait pas parti des entités précédentes.

1.2.1.1.1.2 Reduced Function Device (RFD) Le RFD est un équipement qui possède une implémentation du standard la plus réduite possible. Il est dédié à la réalisation de tâches "simples" : mesure de paramètres environnementaux. Cet équipement étant limité en ressource, le volume de données généré doit être faible. Il est associé à un seul FFD à la fois. Il ne peut pas être connecté à un autre RFD.

Un réseau personnel normalisé IEEE 802.15.4 doit donc inclure au moins un FFD agissant comme coordinateur de PAN, le reste du réseau étant composé de FFD et RFD. Ces diverses entités permettent alors de définir différentes sortes de topologies en fonction des applications.

1.2.1.1.2 Topologies réseau Deux types de topologies réseaux sont définies dans le standard IEEE 802.15.4 : la topologie en étoile (star topology) et la topologie pair-à-

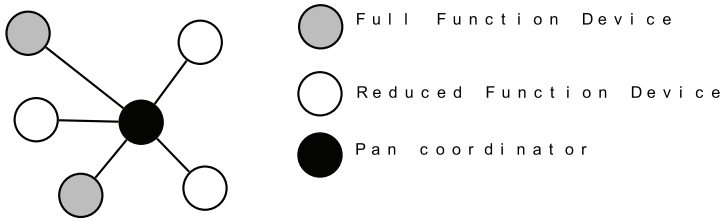

FIG. 1.3 – topologie en étoile

pair (peer-to-peer topology). Le troisième type de topologie, topologie en grappe (the cluster-tree topology), est un cas particulier de topologie pair-à-pair.

1.2.1.1.2.1 Topologie en étoile Dans le cas d'une topologie en étoile (Fig. 1.3), un noeud et un seul est désigné comme coordinateur de réseau personnel. Un FFD peut définir son propre réseau et ainsi devenir son propre coordinateur. Le coordinateur choisit un identifiant de réseau personnel, ce dernier ne doit être utilisé par aucun autre réseau dans l'environnement direct du noeud. Dans ce type d'architecture, les communications sont centralisées. Chaque noeud du réseau (RFD ou FFD) désirant communiquer avec les autres noeuds enverra alors ses données au coordinateur qui les renverra ensuite vers le destinataire adéquat.

Par conséquent, la topologie en étoile ne convient pas au domaine des réseaux de capteurs sans fil. En effet, les noeuds d'un réseau sont, dans la plupart des cas, alimentés par batteries et ont une quantité d'énergie limitée. Un noeud devenant coordinateur prendra donc en charge toutes les communications du réseau et verra son niveau d'énergie drastiquement réduit. Une solution possible à ce problème est la définition d'un coordinateur dynamique basé sur le niveau d'énergie disponible, solution proposé par Heinzelman *et al.* dans le protocole LEACH [40]. Cette solution permet d'augmenter la durée de vie du réseau mais est relativement complexe à mettre en place dans un réseau comportant un grand nombre de noeuds. Le standard IEEE 802.15.4 recommande l'utilisation de la topologie en étoile dans le cadre d'une utilisation personnelle ou domotique.

1.2.1.1.2.2 Topologie pair-à-pair La topologie pair-à-pair (Fig. 1.4) comporte, à l'instar de la topologie en étoile, un coordinateur de réseau personnel. La définition de ce coordinateur est réalisée lors des premières communications et correspond au premier équipement communiquant sur le réseau. Cependant, dans cette topologie, les communications sont décentralisées. Chaque noeud peut communiquer directement avec un autre noeud du réseau sans avoir besoin d'utiliser le coordinateur.

Cette topologie permet une plus grande flexibilité et permet de réduire la consommation d'energie dans le coordinateur. Cependant, elle implique une complexité plus importante dans l'établissement de communications de bout-en-bout et nécessite donc

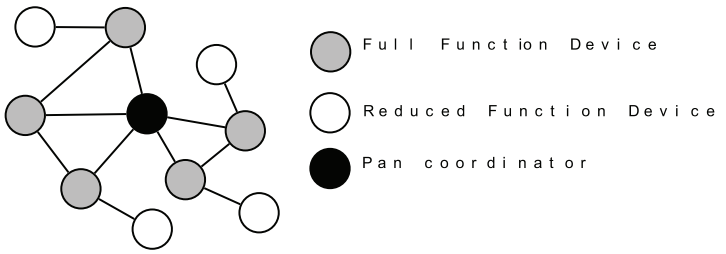

FIG. 1.4 – Topologie pair-à-pair

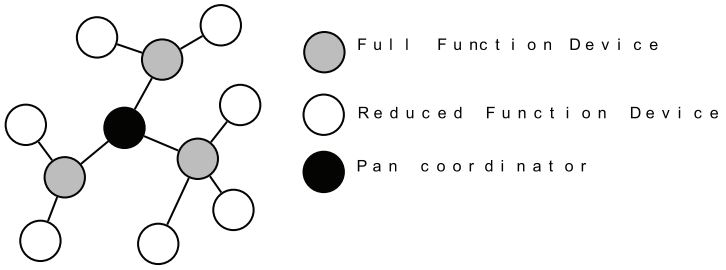

FIG. 1.5 – Topologie en grappe

que chaque noeud soit capable de réaliser ce type de communications. Cette topologie sied parfaitement au réseau de capteurs. L'absence de goulets d'étranglement des communications (*i.e.* coordinateur) permet également d'augmenter la durée de vie du réseau.

1.2.1.1.2.3 Topologie en grappe La topologie en grappe (*i.e. Cluster-Tree*) est un cas particulier d'une topologie pair-à-pair dans lequel la plupart des équipements sont des FDD (Fig. 1.5). Cette topologie comporte les équipements suivants :

- Un coordinateur est nommé en tant que coordinateur du réseau personnel qui gèrera entièrement le réseau.
- Un FFD est désigné comme coordinateur et fournit les services de synchronisation aux autres équipements et aux autres coordinateurs.
- Un RFD peut être connecté à une grappe mais en bout de branche et associé à un et un seul FFD.

1.2.2 Couche physique

La couche physique définit plusieurs fonctionnalités : l'activation de l'émetteur radio, le niveau d'energie du paquet reçu - *Energy Detection (ED)*-, l'indicateur de qualité du lien -*Link Quality Indication (LQI)*-, la sélection du canal, la détection du canal -*clear channel assessment (CCA)*- et la transmission et la réception des paquets à tra-

Bande de fréquences (MHz)	Paramètres de diffusion		Paramètres de données		
	Chip rate (kchip/s)	Modulation	Bit rate (kbps)	Symbol rate (ksymbol/s)	Symbols
868	300	BPSK	20	20	Binaire
915	600	BPSK	40	40	Binaire
868	400	ASK	250	12.5	20-bit PSSS
915	1600	ASK	250	50	5-bit PSSS
868	400	O-QPSK	100	25	16-ary
915	1000	O-QPSK	250	62.5	16-ary
2400	2000	O-QPSK	250	62.5	16-ary

TAB. 1.1 – Paramètres physique de la norme IEEE 802.15.4

vers le médium. Le standard permet de fonctionner dans trois bandes de fréquences distinctes. La technologie utilisée est celle d'étalement de spectre -*Direct Sequence Spread Spectrum (DSSS)*-. Le débit de communications dépend de la bande de fréquences utilisée : 2,4 GHz (250 kbps),915 MHz (40kbps), 868 MHz (20kbps).

Les bandes de fréquences basses permettent d'augmenter la portée des communications et ainsi de couvrir des zones plus grandes. A contrari, une bande de fréquences plus élevée permet un débit plus important, une plus faible latence ainsi qu'un taux d'utilisation plus faible. Ces informations sont résumées dans le tableau 1.1.

Il y a un seul canal dans la bande de fréquences comprise entre 868 et 868.6 MHz, 10 canaux entre 902 et 928.0 MHz et 16 canaux entre 2.4 et 2.4835 GHz (*cf.* Fig 1.6). Le standard fournit un service d'allocation dynamique des canaux, une fonction de scrutation permettant la recherche de balise à travers une liste de canaux, une fonction de détection de niveau d'énergie, une fonction d'indication de qualité de liens ainsi que la commutation de canaux.

La sensibilité à la réception est de -85dBm à 2.4 GHz et de -92 dBm à 868 MHz et 915 MHz. La différence de sensibilité entre ces 2 bandes de fréquence est obtenue par l'utilisation d'un débit plus faible.

1.2.2.1 Niveau d'energie du paquet reçu (ED)

La mesure du niveau d'énergie (*ED*) par le receveur est destinée à la couche réseau pour l'algorithme de sélection du canal. C'est une estimation de la puissance du signal reçu d'un message dans la bande passante d'un canal IEEE 802.15.4. Il est à noter que le standard ne définit pas que l'évaluation du signal reçu d'un message implique que celui-ci soit identifié ou decodé. La valeur est calculée sur un entier de 8 bits de 0x00

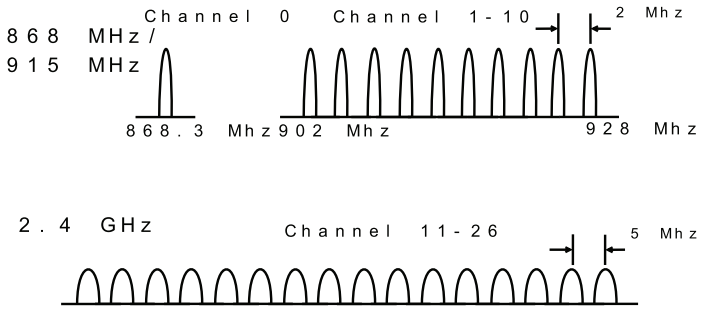

FIG. 1.6 – Bande de fréquences de la norme IEEE 802.15.4

à 0xff. Les valeurs possibles de *ED* s'étalent sur un minimum de 40 dB. Les valeurs obtenues le sont avec une précision de +/− 6d B.

1.2.2.2 Indicateur de qualité du lien (LQI)

La mesure d'indicateur de qualité du lien caractérise l'intensité d'un message reçu. Cette mesure peut être obtenue à partir du niveau d'énergie du paquet reçu (*ED*), du rapport signal à bruit ou d'une combinaison de ces deux valeurs. Le *LQI* peut être utilisé dans les couches supérieures (*e.g.* réseau et/ou transport), mais cela n'est pas précisé par le standard.

1.2.2.3 Détection du canal (CCA)

L'estimation du canal libre *clear channel assessment (CCA)* est effectuée conformément à au moins une des 3 méthodes suivantes :

- Niveau d'énergie supérieur au seuil. CCA signale le medium comme occupé si le niveau d'énergie détecté est au-dessus du seuil *ED* (*ED threshold*).
- Détection de porteuse. CCA signale le medium comme occupé sur détection d'un signal ayant des caractéristiques de modulation et d'étalement conforme à la norme IEEE 802.15.4. Le signal peut donc être en dessous ou au-dessus du seuil *ED*.
- Détection de porteuse avec niveau d'énergie supérieur au seuil. CA signale le medium comme occupé sur detection d'un signal ayant des caractéristiques de modulation et d'étalement conforme à la norme IEEE 802.15.4 et si ce signal a un niveau supérieur au seuil *ED*.

1.2.3 Couche liaison

1.2.3.1 Spécificités de la couche liaison IEEE 802.15.4

La couche MAC 802.15.4 offre certaines similitudes avec la couche MAC de la norme 802.11, telle que CSMA/CA (Carrier Sense Multiple Access / Contention Avoidance) comme méthode d'accès au canal. L'utilisation des périodes de contention (*i.e. Contention Free Period -CFP-* et *Contention Access Period -CAP-*) est également similaire à la norme 802.11. Cependant, les spécifications de la couche MAC prennent en compte les pré-requis des LR-WPANs. Le mécanisme RTS/CTS (*Request-to-Send/Clear-to-Send*) permettant de réduire la probabilité de collision est supprimé. En effet, dans le cas d'un réseau de capteurs où le débit est faible et le volume de données peu important, les messages d'évitement de collisions génèrent un trafic inutile et utilisent des ressources inutilement.

La figure Fig.1.7 présente les différents en-tête existants des paquets 802.15.4.

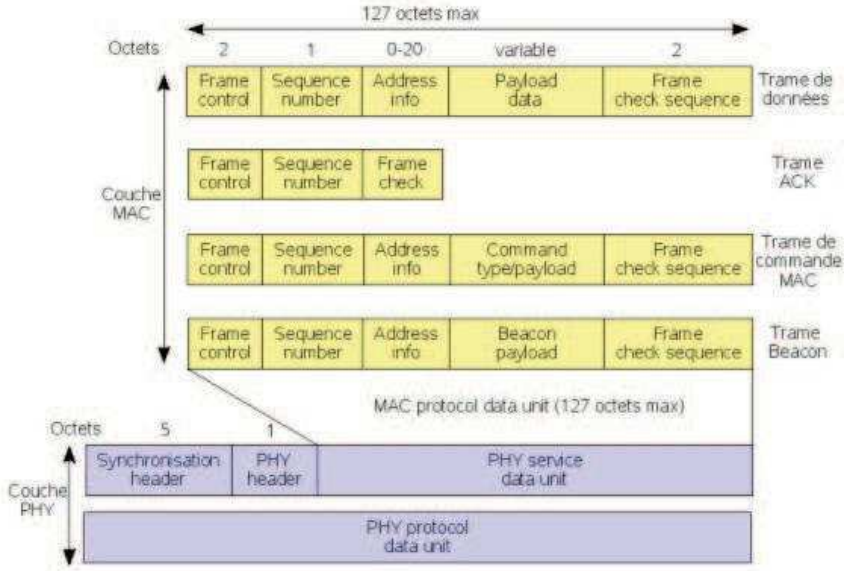

FIG. 1.7 – En-tête des trames 802.15.4 [ref]

Le protocole supporte deux modes opérationnels qui peuvent être définis par le coordinateur de PAN (cf Fig. 1.8) :

1.2.3.1.1 Mode Beacon Les balises sont émises périodiquement par le coordinateur afin d'identifier le réseau et de synchroniser les diverses entités qui y sont rattachées. La trame de balise est la première partie de la supertrame. Le reste de cette

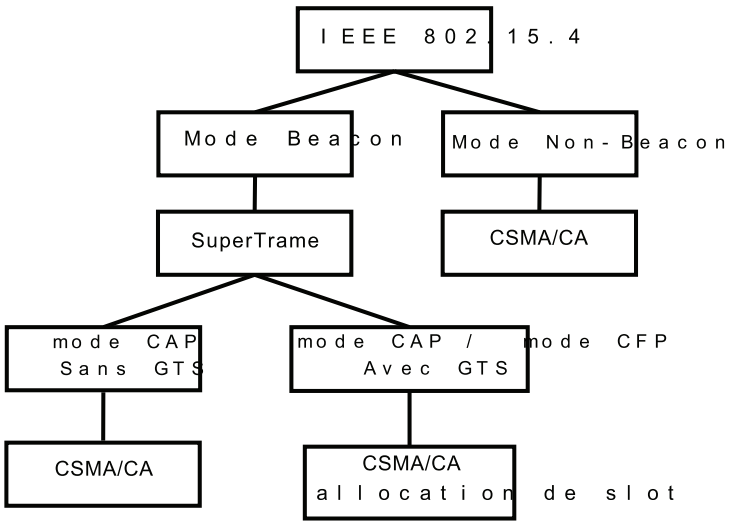

FIG. 1.8 – Les différents modes opérationnels de la couche liaison IEEE 802.15.4

supertrame est composé de l'ensemble des autres messages échangés entre les noeuds et le PAN coordinateur. Les échanges de données directs entre noeuds sont également possibles pendant la durée de la supertrame.

L'intervalle entre chaque balise (BI) définit le temps entre chaque émission de balise. A l'intérieur de ces deux balises, on distingue deux périodes : une période active où les échanges de messages se déroulent et une période inactive où chaque noeud choisit de rentrer en mode *sommeil*, afin d'économiser de l'énergie.

1.2.3.1.2 Mode Non Beacon Dans le mode Non Beacon, il n'y a pas de balise générée par le coordinateur. Les noeuds désirant échanger des données doivent utiliser le mécanisme CSMA/CA pour vérifier l'occupation ou non du canal. La supertrame n'est pas utilisée dans ce mode de fonctionnement.

1.2.3.1.3 La structure Supertrame En mode Beacon, l'usage d'une structure supertrame (Fig.1.9) est requis. Son format est défini par le coordinateur. Elle est délimitée par deux balises et divisée en 16 créneaux de temps égaux. Le premier créneau de chaque supertrame est reservé à la balise. Si un coordinateur décide de ne pas utiliser de supertrame, il lui suffit donc simplement de ne pas émettre de balise.(Fig. 1.9). On distingue alors deux périodes. Pendant la période d'inactivité, le coordinateur entre en mode sommeil. Il n'interagit pas avec le réseau. La période d'activité est divisée en deux périodes : Contention Access Period (CAP) et Contention Free Period (CFP). Pendant la période CAP, les noeuds utilisent la méthode d'accès CSMA/CA pour communiquer. Pendant la période CFP, des creneaux sont alloués et garantis par le coordinateur :

Guaranteed Time Slots (GTS). Le PAN coordinator peut allouer jusqu'à 7 créneaux de temps pendant une période CFP. La durée des deux périodes est décrite à l'aide des variables *Beacon Order* et *SuperFrame Order*. La première variable permet de définir la durée totale de la supertrame (*i.e.* l'intervalle entre deux balises). La valeur *Superframe Order* définit quant à elle, la durée de la période d'activité du noeud. L'intervalle entre deux balises (Beacon Interval) est donc fonction du Beacon Order (BO) :

$$BI = aBaseSuperFrameDuration \cdot 2^{BO} \tag{1.1}$$

ou

- $0 < BO < 14$ et la SuperTrame est ignorée si la valeur de $BO = 15$.
- $aBaseSuperFrameDuration = aBaseSlotDuration \cdot aNumSuperFrameSlots$
- $aBaseSlotDuration = 60$ et $aNumSuperFrameSlots = 16$

La durée de la supertrame, SD, est liée à la valeur *SuperFrameOrder*, SO, selon :

$$\mathrm{SD} = \mathrm{aBaseSuperFrameDuration} \cdot 2^{\mathrm{SO}}, 0 < \mathrm{SO} < 14. \tag{1.2}$$

Si SO = 15, la supertrame ne sera pas utilisée après la balise. La période d'activité de chaque supertrame est divisée en *NumSuperFrameSlots* crénaux de temps dont la durée équivaut à $2^{SO} \cdot aBaseSlotDuration$.

La balise est transmise au début, lors du créneau 0, et sans l'utilisation du CSMA. La période CAP démarre ensuite instantanément. Toutes les trames, exceptées les trames d'acquittement et les trames de données suivant immédiatement les acquittements de données transmises durant le CAP, doivent utiliser le CSMA/CA pour accéder au canal. Une transmission pendant une période CAP doit être completée au minimum un temps IFS avant la fin de cette période. Si cela ne peut être réalisé, la transmission est alors repoussée à la supertrame suivante.

La période CFP, si utilisée, démarre au créneau de temps disponible immédiatement après la période CAP et s'étend jusqu'à la fin de la période d'activité de la supertrame. La longueur de la période CFP est définie par la longueur totale de l'ensemble des créneaux GTS. Contrairement à la période CAP, l'accès au canal se fait sans utiliser le CSMA/CA. Chaque équipement ayant un créneau temporel défini, il n'est pas nécessaire de vérifier la disponibilité du médium pour émettre. A l'instar de la période CAP, une transmission doit être complétée au minimum un temps IFS avant la fin de cette période. La temporisation IFS permet le traitement par la couche physique du paquet reçu. Chaque trame transmise doit donc être suivie d'un temps IFS et sa durée est fonction de la taille de la trame qui vient d'être transmise.

Les réseaux PAN qui ne souhaitent pas utiliser la supertrame dans le mode non-Beacon doivent positionner les valeurs de macBeaconOrder et macSuperFrameOrder à 15. Dans ce cas, le coordinateur ne transmet aucune balise. Toutes les transmissions exceptés les acquittements doivent utiliser la méthode d'accès unslotted CSMA/CA pour accéder au canal. L'utilisation des GTS n'est alors pas possible.

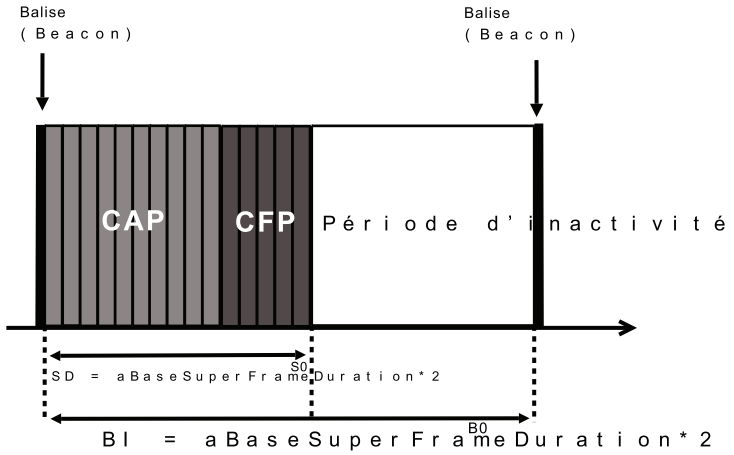

FIG. 1.9 – Structure d'une SuperTrame IEEE 802.15.4

1.2.3.1.4 L'algorithme CSMA/CA Si la supertrame est utilisée, le slotted CSMA/CA est mise en oeuvre (*cf.* Fig.1.10), si elle ne l'est pas, c'est le unslotted CSMA/CA qui sera mise en oeuvre. Dans les deux cas, l'algorithme utilise des périodes de backoff avant une transmission. Dans le cas du slotted CSMA-CA, les bornes de la période de backoff de chaque équipements, sont alignées sur les bornes des créneaux fournis par le coordinateur de PAN (les backoffs de l'ensemble du réseau sont synchronisés). A chaque fois qu'un dispositif souhaite transmettre un paquet, il doit determiner les limites du prochain backoff pour commencer sa période de backoff. Dans le cas de l'utilisation du unslotted CSMA-CA, il n'y a pas de synchronisation des backoffs. La période de backoff commence donc immédiatement.

Afin de mettre en place le mécanisme CSMA/CA, chaque équipement utilise 3 variables : NB, CW et BE.

- NB correspond au nombre de période de backoff réalisé pour la transmission en cours. Il est initialisé à 0 avant chaque nouvelle transmission.
- CW est la longueur de la fenêtre de contention qui définit le nombre de période de backoff sans activité devant être observé avant qu'une transmission puisse être effectuée. Sa valeur est initialisée à 2 avant chaque transmission et réinitialisée à 2 si le canal est occupé. CW est seulement utilisé dans le cas du slotted CSMA-CA.
- BE est le *backoff exponentiel* qui permet de définir la période de backoff qu'un dispositif doit attendre avant de tester le canal. Durant cette période de temps, tous les paquets reçus sont abandonnés, bien que la fonction de réception de l'équipement soit active.

Dans le cas du slotted CSMA/CA, NB, CW et BE sont initialisés et les limites de la prochaine période de backoff sont déterminées. Dans le cas du unslotted CSMA-CA, seul NB et BE sont initialisé. Cette technique d'accès au medium conduit à une

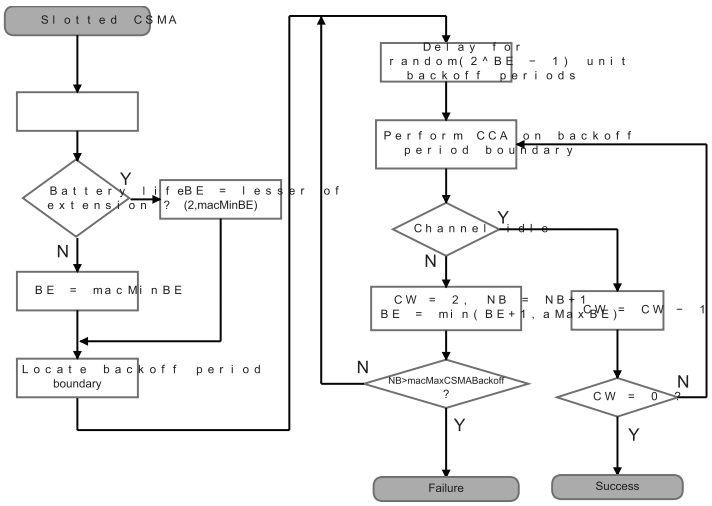

FIG. 1.10 – Slotted CSMA/CA

augmentation de la consommation d'énergie dans le cas où l'on a de longues périodes de backoff. Ce problème apparaît lors de périodes à grand trafic pour éviter les collisions. Cependant, la norme IEEE 802.15.4 supporte le mode d'extension de vie de batterie (BLE : Battery Life Extention), dans lequel les composantes backoff sont limitées entre 0 et 2. Ce dispositif réduit les périodes d'écoute du canal libre pour les applications à faible trafic. Un noeud du réseau doit mettre sa radio en veille pour conserver l'énergie immédiatement après la réception de la trame d'acquittement s'il n'a pas d'autres données à transmettre ou à recevoir.

1.3 couche MAC adaptée aux réseaux de capteurs

Dans cette section, nous présenterons les différents protocoles de la couche MAC développés par la communauté pour les réseaux de capteurs sans fil. Nous commencerons tout d'abord par identifier les caractéristiques particulières des réseaux de capteurs.

1.3.1 Caractéristiques

1.3.1.0.5 Consommation d'énergie C'est une des composantes les plus importantes des réseaux de capteurs. Ces réseaux étant composés d'un très grand nombre de noeuds, l'utilisation d'équipement énergétiquement autonome est quasi-obligatoire. De facto, les solutions envisagées doivent prendre en compte ce paramètre et réduire ainsi la consommation d'énergie. Cela permettra alors de prolonger la durée de vie du

réseau. Un noeud est composé de plusieurs éléments (processeur, unité de stockage, radio ...). Parmi ces éléments, la radio est grande consommatrice d'energie. Son fonctionnement étant directement dépendant de la couche MAC, le protocole utilisé doit donc être adapté.

1.3.1.0.6 Passage à l'échelle et adaptation Les noeuds composant un réseau de capteurs étant, de par leur conception, fragile, la topologie d'un tel réseau peut être modifiée rapidement. Les noeuds peuvent "disparaître" ou se déplacer. Il est donc nécessaire que le protocole MAC soit apte à prendre en compte ces changements de topologies. Ces modifications de topologies peuvent entraîner une explosion du nombre de noeuds du réseau. On peut imaginer que deux réseaux de capteurs disjoints se retrouvent réunis par l'arrivée de nouveaux noeuds entre eux et ne formant alors qu'un seul et même réseau. Le passage à l'échelle est donc également une composante importante des protocoles MAC.

1.3.1.0.7 Temps de latence Le temps de latence est, dans le cas d'un réseau de capteurs, directement lié à l'application. Une application de télésurveillance peut supporter un temps de latence prolongé. En effet, le temps de transport de l'information est beaucoup plus court que le temps d'obtention de la mesure physique en elle-même. Les réseaux de capteurs ont un comportement défini par une grande période d'inactivité suivie par de très courtes périodes d'activité. Un temps de latence supérieur peut être supporté par les noeuds pour la réception des données. Néammoins, ce temps de latence est un paramètre qu'il ne faut pas négliger. En effet, si le temps de latence devient trop important, cela implique que les noeuds doivent rester actifs plus longtemps pour recevoir l'information ou pour attendre l'acquittement d'une donnée envoyée.

1.3.1.0.8 Débit Comme pour le temps de latence, le débit est directement lié à l'application. Dans la plupart des scénarios actuels de déploiement de réseaux de capteurs, le volume de données généré est faible. La majorité des applications privilégie la durée de vie du réseau sur le débit ou le temps de latence. On rappelle que le débit défini par la norme 802.15.4 n'est que de 256 kbits.

1.3.1.0.9 Evitement de collision Comme beaucoup de protocoles de communication basés sur un médium commun, l'évitement de collision est une des tâches principales de la couche MAC. Ce mécanisme est très important car il influe sur les performances des caractéristiques présentées précédemment. La détection de collision est directement liée à la consommation d'énergie, au débit, à la bande passante. De plus, l'utilisation d'un médium sans fil pour communiquer rend l'utilisation de certaines techniques classiques caduque (*e.g.* CSMA/CD). Il est donc nécessaire de définir avec précision un mécanisme d'évitement performant.

1.3.2 Classification des protocoles MAC pour réseaux de capteurs

Langendoen [52] a classifié les protocoles MAC pour réseaux de capteurs selon trois principales catégories : *Random Access*, *Slotted Access*, *Framebased Access*. Nous reprendrons cette classification pour la suite de cette section.

1.3.3 Random Access MAC Layer

C'est la méthode d'accès la plus simple. C'est une solution de type CSMA (*Carrier Sense Multiple Access*).
Les noeuds écoutent le médium de façon périodique afin de déterminer si le canal est libre. Si celui-ci est libre et si le noeud doit envoyer des données, il va alors procéder à l'envoi (nous détaillerons ce processus dans la suite de cette section). S'il n'a pas de données à envoyer et s'il n'y a pas de transmission en cours, il retournera en phase de sommeil. Cela permettra alors d'économiser de l'énergie. En effet, l'écoute passive du médium est une tâche consommant "inutilement" une quantité importante d'énergie. C'est pourquoi les mécanismes s'efforcent de minimiser les temps d'écoute du réseau. Dans ce type de méthode d'accès, on évite d'utiliser des mécanisme d'évitement de collision. La solution RTS/CTS (Request to Send/Clear to Send) au regard de la taille des données, est jugée peu intéressante [78]. Il est plus intéressant de réémettre les données en cas de collision plutôt que de mettre en place un mécanisme pour éviter ces collisions.

1.3.3.1 processus d'envoi

Un des problèmes dans ce type de réseaux se situe lors de la phase d'échange des données. Comme les phases de sommeil sont maximisées, il est nécessaire de mettre en place des mécanismes de synchronisation entre les noeuds. En effet, lorsqu'un noeud se réveille et doit envoyer des données, il doit être sûr que le destinataire des données soit réveillé.

Les noeuds écoutent le médium de façon périodique afin de recevoir et d'envoyer des données. S'il n'y a pas de données à envoyer ou à recevoir, le noeud retournera en sommeil. Ainsi, la consommation d'énergie sera réduite. Polastre *et al.* [78] ont défini un mécanisme permettant à un noeud de s'assurer que le destinataire d'un message est éveillé lors de l'envoi. Ce mécanisme utilise un préambule permettant de signifier aux noeuds voisins lorsque ceux-ci se réveillent, qu'une donnée va être envoyée. Ce préambule est émis sur le canal pendant un temps supérieur à la période de sommeil. Cette période de sommeil doit être fixe et identique dans tous les noeuds du réseaux. Tous les noeuds se réveillant et entendant ce préambule vont alors attendre sa fin et donc rester éveillés. Une fois le préambule terminé, le noeud source enverra le paquet

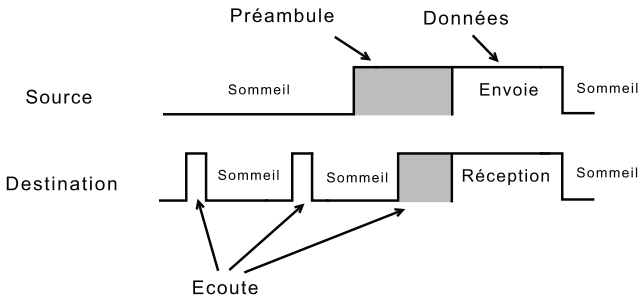

FIG. 1.11 – Utilisation du préambule pour synchroniser la source et la destination

de données qui sera alors reçu par tous les voisins (Fig. 1.11). Chaque noeud devra alors déterminer si le paquet lui est destiné ou non.

L'avantage de ce type de solutions est qu'elles sont distribuées et qu'elles nécessitent peu ou pas d'échanges de messages pour être opérationnelles. A l'inverse, leurs performances s'en trouvent dégradées lorsque le réseau atteint une certaine densité [9, 78]. Chaque noeud agit indépendamment des autres noeuds du réseau. On peut citer comme solutions : B-MAC [78], WiseMAC [30], STEM [83].

1.3.4 Slotted Access MAC Layer

Dans ce type de couche MAC, le temps est divisé en *slot*. Ces *slots* sont défini de façon commune par les noeuds du réseaux. Les phases de réveil sont alors commune à tous les noeuds et les communications en deviennent alors plus aisées. Les noeuds peuvent alors exécuter alors leurs différentes tâches (réceptions et envois de paquets, calculs). Les collisions sont alors gérées grâce à des méthodes de contentions classiques : CSMA-CA. Tous les noeuds se réveillant en même temps, l'envoi et la réception sont simplifiés, il n'est plus nécessaire de synchroniser les noeuds. Cependant le temps de latence est alors proportionnel à la densité du réseau. Ajoutons également que dans ce genre de solution, la synchronisation doit alors être précise et des mécanismes spécifiques doivent donc être développés pour palier à la dérive des horloges.

Dans S-MAC[108], Ye *et al.* ont implémenté, à l'instar de B-MAC, un mécanisme de cycle. Les noeuds du réseau alternent les phases de sommeil et de réveil afin d'économiser de l'énergie (Fig.1.12). S-MAC dispose de mécanismes de synchronisation utilisant des paquets SYNC diffusé à ses voisins. Ils sont envoyés au début de chaque phase active. Ces paquets permettent de connaître l'ordonnancement commun au noeud. Chaque noeud recevant ce SYNC devra alors suivre cet ordonnancement. Si un noeud ne reçoit aucun SYNC, il définit son propre ordonnancement qu'il dif-

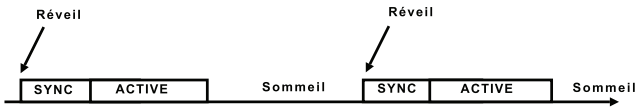

FIG. 1.12 – SMAC

fuse alors pour les futurs noeuds. Ces paquets contiennent également un horodatage permettant à chaque noeud de compenser le problème de dérive d'horloge.

L'utilisation de paquets RTS/CTS permet d'éviter le problème de la station cachée. Il permet également le fonctionnement du mécanisme de sur-écoute, *overhearing*. Ce mécanisme autorise un noeud détectant une transmission RTS ou CTS ne le concernant pas, de passer en phase sommeil. Il se réveillera lorsque la transmission sera terminée. Il utilise pour cela l'information NAV(Network Allocation Vector) inclue dans chaque paquet et qui permet à chaque noeud de connaître le temps d'occupation du canal. Les études ont montré que ce mécanisme permet d'économiser de l'énergie dans un réseau dense [108, 56]. En effet, le trafic augmentant, les messages RTS/CTS sont plus nombreux et les noeuds passent alors plus souvent en phase de sommeil. Dans un réseau à faible densité, les noeuds passeront plus de temps à écouter le canal sans forcément recevoir de données. Un réseau dense aura par contre comme conséquence une grande latence et un débit faible.

On peut également citer les solutions T-MAC [96] ou encore SCP-MAC [109].

1.3.5 FrameBased Access MAC Layer

Ce type de protocole est proche des solutions TDMA (Time Division Multiple Access) : toutes les communications sont ordonnancées. Dans la plupart des solutions, on trouve deux types d'ordonnancement : un ordonnancement des liens et un ordonnancement des sources. Dans le premier cas, chaque paire : source/destination bénéficie d'un slot de temps défini pendant lequel aucun autre noeud ne doit accéder au médium. Dans le cas de l'ordonnancement des sources, chaque source bénéficie d'un slot de temps pour émettre ses données et tous les autres noeuds doivent alors écouter le médium pour vérifier le destinataire du paquet. L'avantage de ces ordonnancements est qu'ils réduisent très fortement les périodes d'écoute inactive. Les collisions sont également très réduites voir inexistantes mais il en résulte un trafic plus important de signalisation pour mettre en place l'ordonnancement.

Rajendran *et al.* ont défini le protocle TRAMA[80], trafic Adaptative Medium Access en supposant que tous les noeuds sont égaux et disposent d'une fonction de synchronisation distribuée. L'occupation du canal se déroule en deux périodes (*cf.* Fig.1.13). Dans la première, l'accès au médium est aléatoire. Chaque noeud peut envoyer des paquets si le médium est libre. Cependant cette période est reservée au trafic de signalisation et d'ordonnancement. Les paquets émis pendant cette période contiennent des

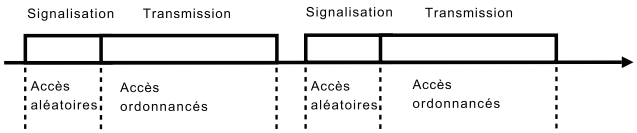

FIG. 1.13 – TRAMA

informations à deux sauts. Pour ces transmissions, TRAMA utilise le protocole *Neighbour Protocol* (NP). Les transmissions sont alors planifiées à partir de la connaissance de la topologie à deux sauts et du trafic à un saut.

Pour cela, l'algorithme *Adaptative Election Algorithm* (AEA) est utilisé. Il permet de déterminer, à partir d'une fonction de hachage basée sur l'identifiant du noeud et le numéro du slot, le noeud qui sera autorisé à envoyer ses données dans ce slot. Enfin le protocole *Schedule Exchange Protocol* (SEP) permet à un noeud, lors de la phase ordonnancée, de communiquer le "planning" de ses futures communications. Ceci permet de s'assurer du réveil des autres noeuds lors de l'envoi des données. Cela permet également à un noeud ne prévoyant pas d'utiliser son slot de temps dans les prochaines communications d'être disponible pour les autres noeuds. Comme pour les protocoles MAC de type slotted, ce genre de collision permet de réduire la consommation d'énergie et de réduire les collisions. Néammoins, on observe, dans ce cas, un temps de latence plus important que dans les protocoles random. De plus, la complexité des algorithmes peut poser problème dans un environnement où les noeuds ont des ressources limitées.

En plus de TRAMA, on peut également citer LMAC[97] ou PEDAMACS[23].

1.4 Communications coopératives dans les réseaux sans fils

Dans un réseau sans fils, la coopération existe à plusieurs échelles. Dans un réseau ad-hoc ou un réseau de capteurs, lorsqu'un noeud envoie un paquet à un destinataire via un ou plusieurs noeuds (routage multi-saut). C'est une forme de coopération qui est mise en place. C'est une technique éprouvée qui permet d'améliorer très fortement les performances du réseau. Dans notre étude, nous nous intéressons à une coopération se situant au niveau de la couche physique et de la couche liaison. Habituellement, les protocoles définis à ce niveau sont directs, c'est à dire que la source envoie directement le paquet au destinataire sans passer par un ou des noeuds relais. Dans le principe de communication coopérative de niveau physique, le noeud source peut utiliser des noeuds relais pour envoyer l'information au noeud destination afin d'assurer une meilleur délivrance du paquet. Dans la figure Fig.1.14, le noeud destination D reçoit le paquet depuis la source S et depuis le relais R.

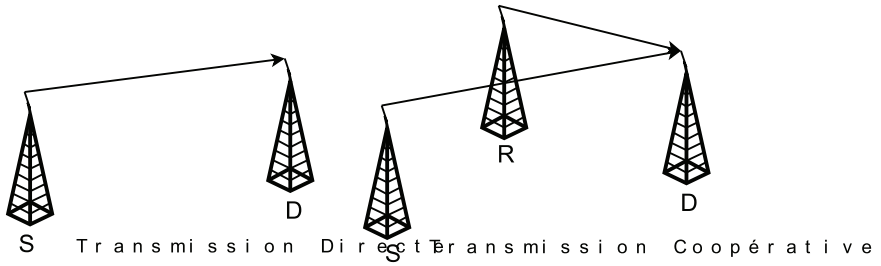

FIG. 1.14 – principe de communication coopérative

Dans ce genre de situation, on est proche des systèmes MIMO classiques où un équipement possède plusieurs antennes radio permettant d'émettre simultanément vers un noeud possédant également plusieurs antennes. Mais les antennes émettrices ne sont plus localisées sur un unique équipement, elle sont distribuées parmis les noeuds et il n'y a qu'une seule antenne réceptrice. Dans notre cas, on parle alors de VMISO : *Virtual Multiple Input Single Output*. Cette technique permet un gain de diversité sans pour autant nécessité un équipement munis de plusieurs antennes.

Pour permettre ce gain de diversité, plusieurs approches de la coopération sont possible : Amplify-and-Forward et Decode-and-Forward.

La première approche consiste à amplifier le signal reçu et de le réémettre vers la destination (*cf.* Fig.1.16). L'implémentation de ce type de coordination est relativement simple et permet une plus grande flexibilité que la précédente solution. Le problème est, qu'en se contentant d'amplifier le signal reçu, on amplifie également le bruit reçu. Il est alors possible que le paquet reçu à destination ne puisse plus être décodé si le bruit est trop important.

Dans la deuxième approche, lorsqu'un paquet est reçu par le noeud relais, il est décodé par celui-ci avant d'être re-émit vers la destination (*cf.* Fig.1.15). Cette solution, bien que plus complexe, permet de meilleures performances. En effet, le noeud relais peut, lors de la réception, assurer la correction du paquet grâce au code correcteur d'erreur. De plus, le paquet étant décodé puis codé de nouveau, il n'y a pas de bruit sur-amplifié comme dans l'approche Amplify-and-Forward. Cependant , le décodage pouvant prendre un certain temps, ceci peut compromettre la coopération ; de même pour l'utilisation du code correcteur d'erreur. L'utilisation de ces mécanismes est donc dépendante des capacités du capteur .Le temps disponible entre la réception et la retransmission du paquet est également à prendre en compte, l'envoi du paquet depuis la source et le relais devant se faire au même instant.

Les signaux reçus sont combinés à la réception selon la loi Maximum Ratio Combining Diversity [16]. Le rapport SNR est alors supérieur à une transmission classique. Celui-ci étant directement lié au taux d'erreur bit (*Bit Error Rate BER*), lorsque le SNR augmente, le taux d'erreur diminue (cf Fig.1.17).

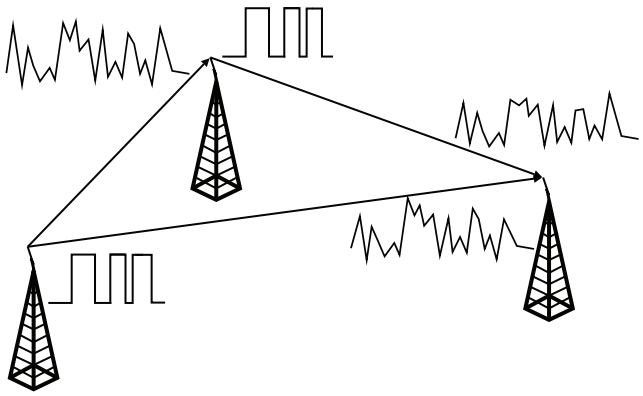

FIG. 1.15 – Decode-and-Forward

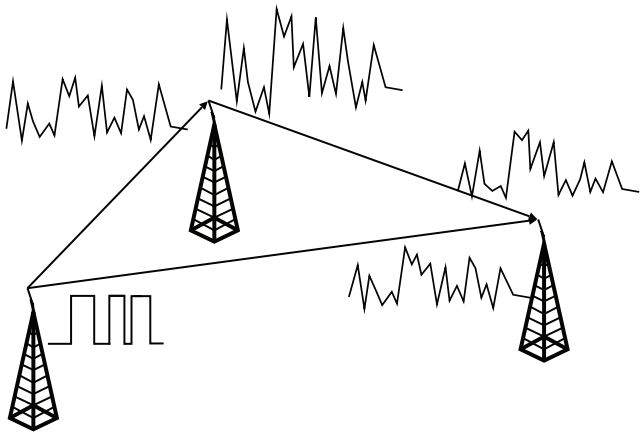

FIG. 1.16 – Amplify-and-Forward

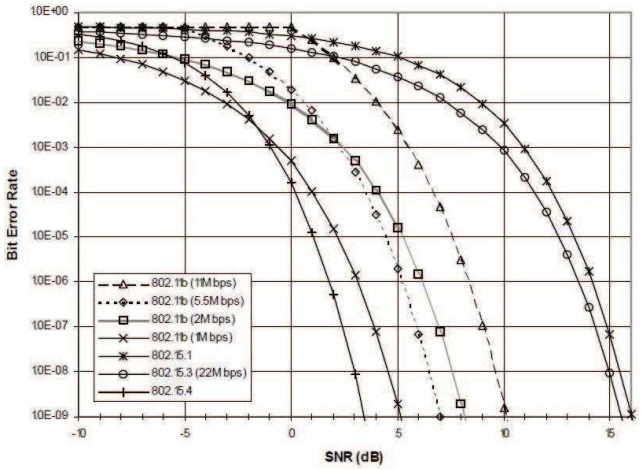

FIG. 1.17 – Bit Error Rate en fonction du SNR(dB) du signal reçu [44]

Laneman *et al.* [50] ont défini un modèle permettant de quantifier la coopération lorsque l'on utilise l'approche Decod-and-Forward avec un codage spatio-temporel. Dans ce cas, on parle de *Space-Time-Coded cooperative diversity*, (diversité coopérative avec codage spatio-temporel). Les noeuds vont alors encoder le message en utilisant le code approprié et le transmettre vers le destinataire. Une transmission d'une source vers une destination se déroule donc en deux étapes dans deux slots de temps différents. Le premier slot de temps permet à la source de transmettre une copie du paquet au noeud relais. Le deuxième slot est dédié à l'envoi du paquet vers la destination depuis la source et le noeud relais.

L'équation Eqn.1.3 représente l'information mutuelle I_{stc} reçue par la destination dans le cas d'une communication coopérative utilisant un codage spatio-temporel et l'approche Decod-and-Forward. $SNR|a_{i,j}|^2$ représente le rapport signal à bruit du signal transmis entre le noeud i et noeud j. m représente le nombre de participants impliqués dans la transmission.

$$I_{stc} = \frac{1}{2}\log\left(1+\frac{2}{m}\mathrm{SNR}|a_{s,d(s)}|^2\right)+\frac{1}{2}\log\left(1+\frac{2}{m}\mathrm{SNR}\sum_r|a_{r,d(s)}|^2\right) \tag{1.3}$$

L'information mutuelle mesure la quantité d'information apportée en moyenne par une réalisation de A sur les probabilités de réalisation de B. Ici, les réalisations A et B sont respectivement la réception d'un paquet depuis la source et la réception du même paquet depuis un noeud relais.

Ces différentes techniques ont été mises en pratique dans différents protocoles.

Ji *et al.* [47] et Lin *et al.* [57] ont proposé différentes solutions définissant un protocole coopératif au niveau MAC. Elles sont basées sur un accès au médium de

type *NDMA Network-assisted Diversity Multiple Access*. Ces solutions permettent de récupérer un paquet grâce à plusieurs paquets reçus ([94]).

Liu *et al.* présentent dans [59], une des premières solutions de protocole MAC coopératif : "CoopMac". Cette solution est basée sur la technologie IEEE 802.11. Deux solutions "CoopMAC I" et "CoopMAC II" sont définies. Dans la première, une trame HTS (Helper ready To Send) est utilisée par le noeud relais pour informer la source et la destination qu'un noeud relais va être utilisé dans la transmission. Le noeud relais peut alors choisir ou non de participer à la communication. Dans l'approche CoopMAC II, la trame HTS n'est pas utilisée, les noeuds utilisent alors l'en-tête de la trame RTS pour établir une transmission coopérative. Dans ce cas, c'est le noeud source qui choisit alors qui sera le noeud relais parmis ses voisins. Le noeud relais doit alors se conformer à ce choix.

Chou *et al.* [20] propose une solution permettant d'établir des communications coopératives dans un réseau sans fil distribué. Ils ont défini des mécanismes permettant de sélectionner un noeud relais parmi les voisins. Ils ont notamment défini une trame RTS spécifique : *Relay-RTS (RRTS)*. Ce RRTS est utilisé conjointement au mécanisme classique RTS/CTS pour informer la source du noeud relais choisi.

Adam *et al.* [][adam pimrc] propose des mécanismes de sélection adaptative de relais. Cette solution implémente des mécanismes utilisant des paquets RTS/CTS et AFR/SFR (Apply For Relay/Select For Relay). Le but de cette solution est de définir le meilleur relais possible. Le premier mécanisme est appelé RSoD (*Relay Selection on Demand*). Cette solution permet aux noeuds relais d'être utilisé sur requete de la destination. Le mécanisme utilise des messages RTS/CTS/AFR/SFR afin d'estimé le PER. Cette estimation permettra de déterminer si la communication a besoin d'un relais. Cela permettra également au noeud destination de définir quel noeud sera le relais de la transmission et d'avertir ce dernier de cette selection. Le second mécanisme, RSer (*Relay Selection with Early Retreat*), permet aux noeuds relais de choisir eux même de participer au processus de sélection du noeud relais.

Dans [28], Dohler *et al.* évaluent les capacités que l'on peut attendre lorsque l'on utilise la coopération dans un réseau de capteurs. Ils établissent que les performances optimales, dans le cas de canaux ergodiques, sont obtenues lorsque la technique de diversité définie par Alamouti [6] est utilisée. La diversité d'Alamouti est obtenue avec 2 antennes réalisant la transmission et une antenne assurant la réception. Dans le cas de canaux non-ergodiques, les performances optimales sont obtenues avec 4 antennes (3/4-rate). Les gains observées sont alors de 5 à 10 dB. Dans notre étude, nos canaux sont ergodiques, nous chercherons donc à établir une coopération basée sur l'utilisation de deux antennes émettrices.

Un canal ergodique est un canal dont le gain est ergodique. La moyenne temporelle est égale à la moyenne spatiale. L'entropie du gain du canal peut-être moyenné, et le calcul de capacité sur le long terme est donc possible. Un canal non-ergodique est un canal dont le gain est une variable aléatoire qui n'est pas foncion du temps. La

moyenne temporelle n'est pas égale à la moyenne spatiale. L'entropie du canal ne peut être moyennée et on ne peut donc pas définir la capacité du canal en utilisant la formule de Shannon par exemple.

Les techniques de communications coopératives permettent d'améliorer les performances des réseaux sans fil, sous condition de définir un noeud relais. Certaines des solutions existantes utilisent des mécanismes basés sur des échanges de trames qui précèdent la transmission elle-même. D'autres font abstraction de ce problème et considérent que le noeud relais est désigné sans pour autant définir le mécanisme permettant de le déterminer.

Dans un réseau filaire classique basé sur des équipements ayant des ressources "illimitées", ces solutions sont parfaitements efficaces. Dans un réseau de capteurs, les ressources sont limitées. Il faut donc réduire au maximum le trafic généré. C'est pourquoi la définition d'un mécanisme optimal permettant de selectionner un noeud relais parmi ses voisins est essentielle.

Nous présenterons dans la suite de ce document, une solution optimisée permettant de définir un noeud relais parmi ses voisins. Cette solution prend en compte les contraintes et les caractéristiques des réseaux de capteurs.

1.5 Robustesse / Fiabilité au niveau réseau

Le besoin ou le manque de fiabilité dans un réseau de capteurs est fortement lié à la spécificité de l'application pour laquelle le réseau de capteurs est utilisé. Considéront un réseau de capteurs déployé pour détecter la présence d'un gaz nocif dans un immeuble habité, avec un point de collecte (également appelé "noeud puits" ou plus couramment "*sink*") ayant la capacité de lancer des requêtes spécifiant quels gaz les capteurs doivent tenter de détecter. Etant donné la nature de l'application, il devient impératif qu'une requête atteigne tous les capteurs de manière fiable. Ainsi, la forme même de fiabilité peut varier d'une application à une autre. Toutefois, tous les réseaux de capteurs déployés pour pourvoir à une application critique requièrent des mécanismes pour assurer le transport d'information depuis le sink vers les capteurs. Différents types de transport fiable peuvent être requis dans un réseau de capteurs en fonction de la nature des messages et de l'étendue des capteurs visés. On distingue 2 types de fiabilité basée sur la taille des messages et le nombre de paquet nécessaires devant être utilisé pour la transmission :

- **Transport d'un paquet seul** : on peut s'attendre à ce que la plupart des requêtes dans un réseau de capteurs soient suffisamment petites pour tenir dans un simple paquet de donnée. S'assurer de la bonne réception d'un simple paquet est plus couteux que s'il s'agissait d'une donnée émise sur plusieurs paquets. Le récepteur ne peut utiliser des mécanismes de type NACK (negative acknowlegment). Dans le cas de paquet unique, il faudrait acquitter chaque paquet. Dans

le cas de train de paquets, on peut détecter la perte d'un paquet par l'utilisation de séquencement des paquets.

- **Transport de plusieurs paquets** : Ce genre de transport de message est d'autant plus intéressant grâce à la redondance inhérente présente sous la forme de paquets multiples. Un capteur ne recevant qu'une partie du message peut participer activement à l'obtention des autres paquets.

Le nombre de capteurs visé constitue une autre distinction dans les types de fiabilité :

- Message délivré à tous les capteurs du réseau.
- Message délivré à tous les capteurs d'une zone restreinte.

Les caractéristiques d'un réseau de capteurs qui doivent être pris en compte pour développer une solution au problème de fiabilité de la délivrance des messages sont définis par Park *et al.* [73]) :

- **Passage à l'échelle** : La taille d'un réseau de capteurs est supérieur à un réseau ad hoc ordinaire. Ainsi le problème issu du fait que l'on n'utilise aucune forme de coordonnées globales est hautement critique.
- **Densité** : Dans un environnement de capteurs, on peut s'attendre à ce que la densité de noeuds soit très élevée et que les problèmes de diffusion soient sévères et requièrent des mécanismes de fiabilité robuste.
- **Rareté des ressources** : Les réseaux de capteurs, comme les réseaux ad hoc, sont caractérisés par une bande passante limitée et la puissance de la batterie. Donc, les mécanismes de fiabilité doivent être hautement efficaces.
- **Temps de latence** : Une requête a besoin d'être délivrée de façon fiable mais également dans un délai borné.

Les caractéristiques d'un réseau de capteurs permettent donc de prendre des décisions clefs pour la mise en place de la fiabilité :

- **ACK vs NACK** : Utiliser une méthode de fiabilité basée sur des acquittements (ACK) entraineraît le traditionnel problème d'implosion des ACKs, lequel peut être spécialement sévère dans un réseau de capteurs vu son étendue. Un procédé d'acquittement négatif (NACK) est préférable [89]. Toutefois, un procédé basé sur des NACKs doit toujours répondre au problème éventuel de l'implosion des NACKs dans un voisinage local, et doit également être combiné à d'autres mécanismes pour assurer la délivrance des paquets uniques.
- **Récupération Locale contre Non-Locale** : Les réseaux de capteurs pouvant êtres très grands et comporter plusieurs milliers de noeuds, la récupération locale est de loin meilleure.
- **Serveur de Récupération désigné ou non-désigné** : L'avantage des serveurs désignés est l'amélioration potentielle des performances qui peuvent être gagnées par une désignation appropriée [55, 74]. Une approche par serveur non-désigné, d'un autre coté, peut résulter en de larges en-têtes dûs au manque de coordination.

- **Désignation Dynamique contre Statique** : Etant donné la nature dynamique des réseaux, il est nécessaire que la désignation des serveurs de récupération soit faite au plus tôt possible après l'envoi du message.
- **Propagation hors-séquence ou en-séquence** : Un procédé basé sur des NACKs couplé à de la propagation des paquets d'un message dans un ordre aléatoire peut déclencher des messages NACKs non nécessaires transmis par tous les serveurs en aval [74, 102]. D'un autre coté, ne propager des paquets que dans le bon ordre peut entraîner le gaspillage de la bande passante.

Dans la suite de cette section, nous nous attarderons sur la problématique de fiabilité au niveau réseau en présentant diverses solutions existantes actuellement.

1.5.1 Le protocole MOR

Le Multipath On-demand Routing protocol (MOR) [14] est un protocole qui permet aux différents noeuds d'un réseau de capteurs sans fil de se connecter ensemble. C'est un protocole de routage ad-hoc qui est réactif ou, *on-demand*, c'est-à-dire qu'il établit des routes que lorsque cela est nécessaire. L'avantage de cette approche est évidente si quelques routes seulement sont nécessaires, puisque les trafic dû au routage est moindre comparé à l'approche proactive où l'on établit même les routes qui ne sont pas nécessaires. Un défaut de la création de routes *on-demand* est les temps de créations des routes nécessaires. Comme d'autres protocoles du même type, MOR s'appuie sur des flux de diffusion, également connus sous le nom d'inondation du réseau, pour découvrir une route vers une destination quand une telle route n'existe pas. Ce type de protocole sera présenté dans la suite du présent document.

Dès lors, la première optimisation introduite par MOR est l'inondation du réseau par la station de base quand le réseau se met en marche, puis de manière périodique. Par ce biais une seule inondation de réseau permet d'établir et de maintenir des routes depuis chaque noeud du réseau vers la station de base. Ce qui doit être comparé avec la quantité d'inondations si chaque noeud initie la procédure pour avoir un chemin vers la station de base. Une autre optimisation de MOR est l'utilisation de l'unicast autant que possible. En effet, si la diffusion est utile dans les inondations de réseau, il vaut mieux utiliser de l'unicast pour les paquets de données pour permettre à chaque noeud qui ne serait pas le suivant dans la transmission de s'éteindre et économiser son énergie. Enfin la principale optimisation de MOR qui le distingue des autres protocoles de routage ad hoc est l'enregistrement de plusieurs chemins de même coût vers une même destination, alors que les autres n'en gardent qu'une, d'où la nature Multi-path de MOR. Le fait d'utiliser plusieurs chemins vers une destination permet de :

- Distribuer la charge de manière équitable
- Utiliser la bande passante du réseau plus efficacement
- Faire en sorte que la consommation d'énergie se fasse de manière uniforme

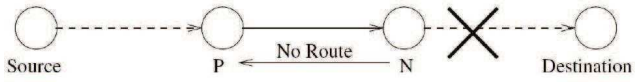

FIG. 1.18 – Annulation d'une route [14]

1.5.1.1 La couche de fiabilité de MOR

Formellement, la couche de fiabilité de MOR est un protocole de retransmission de la couche liaison de données. Quand MOR souhaite envoyer un paquet de manière unicast, il consulte sa table de routage. Toutes les routes à travers un certain noeud vers une certaine destination ont le même coût, et chacune a un saut suivant différent. Le noeud choisit la route qui fut utilisée la moins récemment et transmet le paquet vers le noeud suivant. La couche de fiabilité surveille la transmission à l'affût d'erreurs. Si un paquet est envoyé un nombre déterminé de fois (3 par défaut) vers le même noeud et que celui-ci n'acquitte pas la réception, alors la couche de fiabilité regardera la table de routage de MOR à la recherche d'une autre route vers la même destination. Toutes les routes seront essayées avant de détruire le paquet. De plus, si un noeud échoue à transmettre des paquets trois fois de suites, toutes les routes à travers ce noeud seront effacées. Si le problème est réglé, le noeud ne sera pas immédiatement réintégré dans les tables de routage de ses voisins, mais seulement s'il y a une nouvelle requête de découverte de chemins ou si le noeud veut lui même transmettre des données.

1.5.1.2 Gestion des routes actives

Quand un noeud N efface sa dernière route vers une certaine destination D, il n'y a alors plus aucune raison pour les noeuds en amont de continuer à utiliser N comme noeud suivant pour leurs communisations avec D. Pour communiquer cette information vers les noeuds en amont, un noeud N qui détruit un paquet qu'il a reçu par manque de route disponible envoi un message *No Route* au noeud précédent P qui lui a transmis le paquet. P peut alors lui aussi effacer la route qui passe par N. Si par contre P possède une route alternative vers D, il peut en informer N pour qu'il sache qu'il peut utiliser P comme moyen d'accès à D (Fig. 1.18).

1.5.1.3 Avantages de la couche de fiabilité et de la gestion des routes actives

La combinaison de ces deux mécanismes fournit une fiabilité accrue à un coût énergétique moindre. Les routes sont préservées contre des problèmes transitoires comme des congestions localisées, elles sont rétablies rapidement quand c'est possible, et l'utilisations d'inondations est réduite. Les retransmissions de la couche liaison de données peuvent être bien plus rapides que les retransmissions de bout en bout, alors

le mécanisme de contrôle de congestion de TCP (si c'est lui qui tourne au dessus de MOR) sera rarement invoqué. Quand, d'un autre côté, tout le réseau est congestionné, la couche de fiabilité échouera à transmettre les paquets et le mécanisme de contrôle des congestions de TCP peut prendre la main, régissant correctement à la congestion en réduisant le débit.

1.5.2 REAR : Reliable Energy Aware Routing

Le protocole REAR [39] se donne pour objectif d'assurer la fiabilité de l'acheminement des données à un coût faible en énergie, et cela en minimisant les retransmissions. C'est un protocole de routage fiable qui tient compte de l'énergie disponible, et qui croise les couches réseau et transport. Contrairement à d'autres protocoles qui essaient seulement de trouver un chemin aux frais énergétiques minimums (en prenant le risque de réutiliser ce chemin si souvent que les noeuds de celui-ci dissipent leur énergie plus rapidement que les autres), REAR permet de trouver un chemin avec "suffisamment" d'énergie (et ce ne sera pas forcément le chemin le plus court, mais celui qui aura suffisamment d'énergie tout en étant assez court entre la destination et la source). REAR fournit un environnement de transmission robuste, basé sur un routage respectueux de l'énergie et sur des mécanismes de réservation de l'énergie dans la couche réseau, et qui fournit une transmission fiable à la couche transport.

L'algorithme est constitué de :

- **Path Discovery**, dont Service Path Discovery (SPD) — découverte du chemin principal — et Backup Path Discovery (BPD) — découverte du chemin secondaire
- **Energy Reservation**, dont Service Path Reservation (SPR) — Reservation de l'énergie pour le chemin principal — et Backup Path Reservation (BPR) — réservation de l'énergie pour le chemin secondaire
- **Reserved Energy Release** (RER) — libération de l'énergie réservée

Considérons un réseau de capteurs composé de noeuds ou chacun de ces noeuds peut réaliser une tache (*e.g.* mesure d'une caractéristique de l'environnement) lorsque le sink du réseau lui demande. Quand un sink reçoit une requête depuis un terminal de contrôle, il regarde dans sa table de routage s'il existe une route vers la source devant realisé cette tache. S'il n'existe aucune route, le sink lance alors une requête SPD. La requête est diffusée dans le réseau jusqu'à ce qu'elle atteigne la source visée. Une fois que la source accède aux informations issues de la "première" requête reçue, une requête de réservation d'énergie pour chemin principal (SPRQ : Service-Path Reservation reQuest) sera générée. La requête SPRQ est envoyée de manière unicast vers le sink, le long du chemin emprunté par la première requête. Chacun des noeuds intermédiaires du chemin va réserver une partie de son énergie pour ce chemin (Fig.1.19). La procédure de découverte de chemin principal (SPD) s'achève quand le sink récupère le message SPRQ. Ensuite, toutes les transmissions entre le sink et la source seront

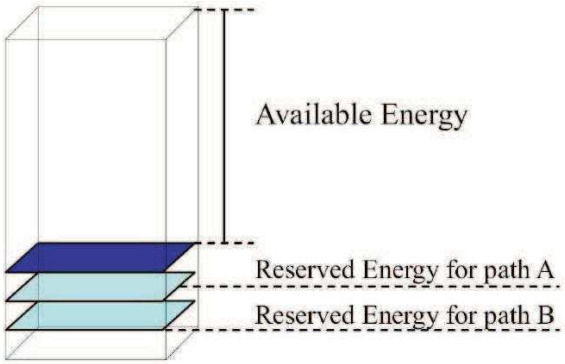

FIG. 1.19 – Partition de l'Energie dans les Noeuds Intermédiaires [39]

unicast et fiables. Au même instant, le sink va lancer une requête de découverte de chemin secondaire (BPD) vers la même source. Seul les noeuds qui ne sont pas sur le chemin principal pourront diffuser la requête. Ainsi, un chemin tout à fait disjoint sera établi et réservé (BPRQ). La libération de l'énergie réservée sera initiée quand un chemin sera rompu.

REAR fournit une transmission fiable, une fois que le chemin est établi entre le sink et la source. Chaque noeud intermédiaire gardera en mémoire les données en cours d'acheminement jusqu'à ce qu'il reçoive un acquittement depuis le noeud suivant. Etant donné la taille des mémoires des capteurs, la source n'est pas en mesure de maintenir dans son cache l'intégralité des données qu'il transmet, en vu d'une éventuelle retransmission. C'est pourquoi, quand une panne est détectée sur un chemin, les noeuds intermédiaires renverront les segments du paquet qui n'aurons pas atteint le sink à la source. Quand un chemin se brise, un rapport d'erreur est généré et diffusé aux deux terminaux. L'ancien chemin sera retiré des tables de la source et du sink. L'énergie réservée sur le chemin brisé sera relâchée, et toutes les transmissions entre le sink et la source basculeront sur un autre chemin disponible, et cela sans discontinuité (car la source indique une route à chacun des segments). Sans le backup-path, la source aurait dû garder en mémoire l'intégralité des données jusqu'à la découverte d'un nouveau chemin. Or, la taille des mémoires étant petite, si la source continue à générer des données, la majorité des données seront perdues. La protection apportée par le chemin secondaire préétabli réduit le délai d'acheminement des paquets et maintient la mémoire utilisée des capteurs à un niveau acceptable. Nous avons vu que pour découvrir un chemin, chaque noeud diffusait la requête du sink, et la première requête qui atteindrait la source définirait le chemin. Or par la nature même de la diffusion, cette méthode ne permet que de trouver le chemin le plus court (donc qui consommera globalement le moins d'énergie) et non le meilleur (*i.e.* qui disperserait au mieux les dépenses d'énergie entre les noeuds). Pour pallier à ce manque, il suffit aux noeuds de combiner la vitesse

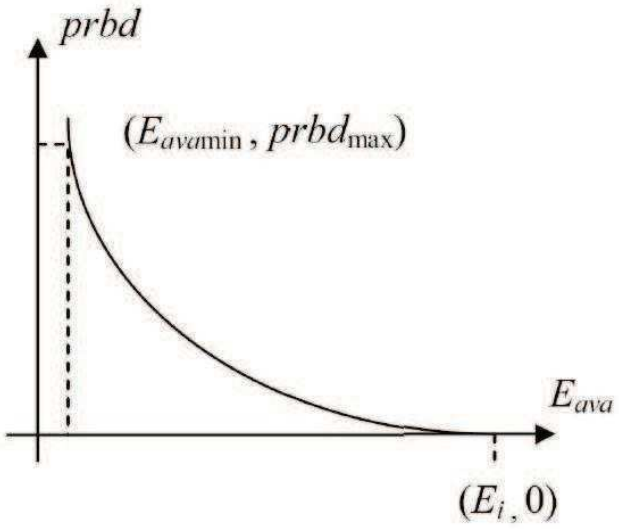

FIG. 1.20 – Représentation du *prbd* [39]

de diffusion avec l'énergie dont ils disposent. Cela signifie que REAR introduit un délai de propagation des requêtes de recherche de chemin (*prbd* : path-request broadcast delay).

En d'autre terme, quand un noeud reçoit une requête de recherche de chemin, il ne le diffuse pas immédiatement vers ses voisins. Plusieurs choses doivent être faites au préalable. Si l'énergie disponible est inférieure à celle nécessaire (par exemple, deux fois l'énergie nécessaire à la transmission d'un paquet), cela indique au noeud qu'il ne peut se permettre de participer à de nouvelles transmissions. Le noeud détruit tout simplement le paquet. Dans le cas contraire, il calcule le *prbd*, dont la valeur indique implicitement le niveau d'énergie disponible dans le noeud. Le noeud maintient la requête durant un délai *prbd*, puis il diffuse la requête vers ses voisins. En utilisant cette méthode, un noeud faible sera écarté d'un éventuel chemin, le *prbd* étant inversement proportionnel au niveau d'énergie disponible (Fig. 1.20). Quand la requête atteint la source, cette dernière récupère les informations sur le chemin emprunté par le premier des paquets reçus, et ce chemin est candidat à la place du chemin principal entre le sink et la source (les communications sont bidirectionnelles).

1.5.3 Protocoles de routage dans les réseaux Ad-Hoc mobiles

Actuellement, la communauté des réseaux de capteurs, via des groupes de travail tel que 6LowPAN [67] s'intéresse au problématique d'adressage dans les réseaux de capteurs. La problématique de communication entre réseaux de capteurs et réseaux

classiques (*e.g* IP) a faire ressurgir l'intérêt d'un protocole d'adressage commun à tous ces réseaux. L'interconnexion des réseaux via des passerelles est une solution possible mais qui reste complexe à mettre en oeuvre. En utilisant un adressage commun aux deux réseaux, les communications pourraient se faire par l'utilisation d'un protocole d'adressage commun à ces deux réseaux. Chaque capteur serait "visible" sans passer par l'intermédiaire d'une plate-forme. L'utilisation d'IPv6, grâce à sa très large plage d'adresses disponibles, répond à certaines contraintes des réseaux de capteurs. L'émergence d'IPv6 a posé la question des protocoles de routages à utiliser avec cette adressage. 6LowPAN définie notamment des mécanismes permettant de réduire la taille de l'entête afin qu'IPv6 puisse être utilisé sur des réseaux de capteurs de type 802.15.4. La couche physique du standard ne supporte que des paquets de 127 octets maximum. L'entête classique de IPv6 est codé sur 40 octets et représente donc près du tiers d'un paquet. L'utilisation des informations contenue dans l'entête MAC pour définir l'entête IPv6, ainsi que de le développement de mécanisme spécifique a permis de réduire la taille de cette entête à deux octets. L'utilisation de 6LowPAN permet donc la communication entre terminaux 802.15.4 mais également entre terminaux 802.15.4 et terminaux non-802.15.4.

Le groupe de travail Roll (*Routing Over Low power and Lossy networks*)[99] étudie les divers protocoles de routages existants et développe une architecture de routages spécifique aux réseaux de capteurs permettant d'utiliser un protocole d'adressage IP. Ce groupe de travail prend appui sur les faiblesses et les qualités des protocoles de routages existants, notamment ceux développés pour les réseaux mobiles ad-hoc.

Un réseau mobile ad-hoc (MANET) est un réseau composé d'équipements dont la principale caractéristiques est d'être dépourvu d'infrastructure et d'une architecture centralisée. Dans ce type de topologie, chaque noeud agit comme un routeur potentiel. Les réseaux de capteurs partagent certaines caractéristiques avec les MANETs, c'est pourquoi nous présentons ici les protocoles de routages développés pour ces réseaux.

Les réseaux MANETs sont également caractérisés par une topologies dynamique et aléatoire. Les algorithmes de routages doivent garder des tables de routage "raisonnablement" petites, au regard des faibles capacités des dispositifs. Ils doivent également être capables de réagir aux modifications de topologies afin de trouver la meilleure route vers la destination. Les protocoles de routages sont divisés en deux principales catégories : protocoles proactifs et protocoles réactifs.

La littérature étant très complète sur ce sujet, nous nous contenterons de citer les protocoles proactifs les plus connus : OLSR [21], Destination-Sequenced Distance-Vector (*DSDV*)[76], Topology Dissemination Based on Reverse Path Forwarding (*TBRPF*)[71] et Direction Forward Routing (*DFR*) [53]. Pour notre étude nous nous intéressons plus particulièrement aux protocoles réactifs.

L'idée développée par les algorithmes de routages réactifs est la construction des routes à la demande, c'est-à-dire que l'on détermine la route vers la destination seulement lorsque c'est nécessaire. Cela permet de réduire la consommation de bande pas-

sante et d'énergie. Cependant, ce type de solutions induit un temps de latence dans la transmission des paquets. La route n'étant pas disponible immédiatement, il faut d'abord inonder le réseau par des requêtes pour trouver le chemin vers la destination. Cela induit un nouveau problème qui est l'inondation du réseau par des réquetes de détermination de routes. Cette inondation est grande consommatrice de bande passante. Elle a néammoins l'avantage de ne se faire que si nécessaire, contrairement aux solutions proactives. Des solutions propose des mécanismes permettant de limiter le nombre d'inondations [14]. Ils existent plusieurs solutions tels que Dynamic Source Routing (DSR), Ad Hoc On-Demand Distance Vector Routing ($AODV$) et Dynamic Mobile On Demand (DYMO). Nous avons basé une partie de notre travail sur les algorithmes de routage réactifs. Nous présenterons donc succintement dans la section suivante ce type de protocoles.

1.5.3.1 Les protocoles de routages réactifs

Nous présenterons dans cette section, divers solutions reposant sur l'utilisation de routes à la demande.

1.5.3.1.1 Dynamic Source Routing (DSR) DSR est un protocole de routages réactifs qui suppose que chaque noeud du réseau agit comme un routeur. Il suppose également que les communications entre deux noeuds ne soient pas forcément bi-directionnelles. Le routage utilisé est dit routage à la source par opposition aux routages saut par saut. La route à utiliser est choisie par la source lors de l'envoi du paquet et non par les noeuds à chaque saut.

Les paquets émis contiennent les informations de routages, ce qui signifie que les noeuds intermédiaires n'ont pas besoin de calculer la route à utiliser lorsqu'ils reçoivent un paquet. Le protocole DSR est décomposé en deux phases : Découverte de route et maintenance de route. La découverte de route est le mécanisme permettant de définir la route vers une destination donnée. La source va inonder le réseau avec des requêtes de route et attendre de recevoir une réponse lui indiquant la route à utiliser pour atteindre le destinataire. La maintenance de route est nécessaire lorsqu'un noeud détecte qu'un lien est indisponible ou qu'un changement de topologie a rendu une route inutilisable. Ce noeud notifie la source de cette modification en envoyant un message d'erreur (Route Error). L'expéditeur peut alors choisir de recommencer le processus de découverte de route et d'envoyer le paquet sur la nouvelle route ainsi calculée.

1.5.3.1.2 Ad-Hoc On-Demand Distance Vector Routing (AODV) $AODV$ est une combinaison de DSR et de DSDV. Les fonctionnalités de découverte de routes et de maintenance de routes sont les mêmes que DSR. En plus de ces mécanismes, $AODV$ emprunte à DSDV le routage de proche en proche, le mécanisme de séquencement et les trames de balises périodiques. Les communications entre noeuds sont supposées

bi-directionelles. L'utilisation du séquencement permet à *AODV* de ne pas conserver les routes trop anciennes, leurs numéros de séquence fonctionnant à la façon d'un horodatage.

1.5.3.1.2.1 Fonctionnalités Quand un noeud cherche le chemin vers une destination, il inonde le réseau avec un paquet *route request (RREQ)*. Chaque noeud recevant ce paquet ajoute son propre identifiant au chemin de retour contenu dans la requête. Il transmet ensuite cette dernière à ses voisins. Quand la requête atteint la destination ou un noeud connaissant une route vers la destination, une réponse est générée. Cette réponse se présente sous la forme d'un paquet *Route Response (RREP)* contenant le chemin défini dans le *RREQ* et le chemin vers la destination. Le mécanisme de maintenance de routes est effectué en local par l'envoi périodique de messages *hello*. Un lien entre deux voisins est considéré rompu si un message hello n'est pas reçu au bout de trois périodes d'envoi. Si une rupture de lien est observée par un noeud, celui-ci va alors informer les noeuds utilisant la route ainsi affectée. Ces noeuds pourront alors mettre à jour leur table de routage et avertir à leur tour leurs voisins de l'indisponibilité de la route.

1.5.3.1.2.2 Utilisation des ressources Les entrées des tables de routage sont fixes et composées de trois élements : saut suivant, durée de vie et statut (actif ou inactif). Cela signifie que la taille de la table de routage peut théoriquement atteindre *n-1* entrées, ou *n* représente le nombre total de noeuds dans le réseau. Il est donc théoriquement possible de saturer la mémoire du noeud et le rendre inopérant. Cependant, *AODV* étant un protocole à la demande, la table de routage ne se remplit qu'à l'envoi d'un message et ne se vide que si la route n'est plus valide. Il est donc fortement improbable qu'une table de routage contiennent tous les noeuds du réseau.

La consommmation d'énergie induit par le processeur est réduite avec *AODV*. Il y a peu de calculs à faire pour déterminer la route à utiliser. L'*overhead* introduit est dû à la découverte de routes (*RREQ* et *RREP*).

1.5.3.1.3 Evolutions actuelles de AODV AODV a bénéficié des critiques de la communauté réseaux et s'appelle maintenant DYMO [42]. Il existe également une implémentation pour les réseaux de capteurs : Tymo [95]. Cette implémentation fonctionne actuellement sur la plate-forme TinyOS [93].

Première partie

Réseaux de capteurs pour l'assistance aux personnes

Les réseaux de capteurs pour l'aide aux personnes permettent d'apporter une assistance aux personnes mais également dans le cas d'application de télémedecine de surveiller les données médicales des patients en milieu hospitalier ou chez eux Ce type de réseaux comporte de nombreuses contraintes liées à l'environnement physique mais également au contexte sensible logiquement induit par la composante médicale de ces réseaux.

Dans ce chapitre, nous présenterons tout d'abord une plate-forme de communications développée et implémentée pour le compte de la Régie Autonome des Transports Parisiens (RATP). Cette plate-forme permet la communication entre équipement mobiles au sein d'une gare de métro. Elle a été déployée à la gare de métro "Bastille" et nous a permis d'observer le comportement des réseaux personnels sans fils (WPAN) en environnement clos.

La réduction de la consommation d'énergie dans les réseaux de capteurs pour la médecine a fait l'objet d'étude et nous y consacreront donc une section de ce chapitre. celle-ci se concentrent sur l'aspect transmission de données de ces réseaux et ont permis de mettre en avant une réduction de la bande passante utilisée et une amélioration de la fiabilité des communications.

Nous exposerons enfin une architecture de sécurité permettant la création d'un tunnel de communications entre une entité du réseau de capteurs et une entité située sur un autre réseau technologiquement disjoint (*e.g.* le terminal de surveillance). Cette architecture permet d'assurer une confidentialité de bout-en-bout des données circulant sur le réseau de capteurs et sur le réseau du terminal de surveillance.

Chapitre 2

Plateforme de communication pour WPAN

Dans cette section, nous présenterons une architecture développée dans le cadre d'un projet mené avec la Régie Autonome des Transports Parisiens (RATP). Ce projet avait pour but de fournir un outil de communication aux usagers des transports en commun. Cet outil devait permettre aux personnes de pouvoir communiquer entre elles, communiquer avec le personnel de la RATP pour l'assistance et obtenir des informations sur l'état du trafic. Enfin, l'implantation de la plate-forme devait se faire dans une gare du métro parisien. De nombreux points de cette architecture (WPAN, environnement clos, mobilité des personnes ...) de communication sont semblable aux caractéristiques d'architectures existantes dans les réseaux de capteurs pour la médecine. Nous détaillerons dans la suite, les différents aspects de cette plate-forme.

2.1 Aspects technologiques

Le développement de cette plate-forme avait pour principale contrainte de pouvoir être utilisée par un maximum de personnes sans nécessité d'investir dans des dispositifs de communications coûteux. Il a donc été décidé d'utiliser la technologie Bluetooth qui équipe de nombreux téléphones mobiles actuels. Cette technologie est définie par la norme IEEE 802.15.1 et IEEE 802.15.2. Les équipements Bluetooth ont une portée de communications relativement faible (100m pour la norme 802.15.1). Dans un environnement clos tel qu'une gare de métro, cette portée est suffisante pour assurer une communication entre deux équipements situés dans une même zone. Mais, elle ne permet pas à deux dispositifs situés dans deux couloirs différents de communiquer entre eux. Il a donc été nécessaire d'utiliser des antennes relais pour transférer les communications issues des équipements Bluetooth.

Le choix s'est porté sur des bornes Bluetooth de la société Bluegiga [15]. Ces bornes possèdent un noyau linux et 3 antennes de communications disjointes. Chacune de ces

antennes fonctionnent indépendamment des autres. Le point d'accès peut donc émettre et recevoir des données au même instant. En outre le point d'accès est capable de gérer 21 connexions simultanément. Nous avons donc choisi d'assigner aux antennes et aux connexions des tâches particulières. 7 connexions étaient réservées à la détection des personnes, 7 connexions étaient dédiées à l'émission de paquets et les 7 dernières connexions permettaient de recevoir des données. Elles sont également munie de divers interfaces de communications (Ethernet, wifi, etc..). L'architecture du système d'exploitation (cf Fig.2.1) permet de développer des applications capable de recevoir, analyser et transmettre des paquets de données sur les différentes interfaces possibles. Les points d'accès sont reliées entre eux et au serveur par le biais de liaison ethernet et de liaison utilisant la technologie CPL.

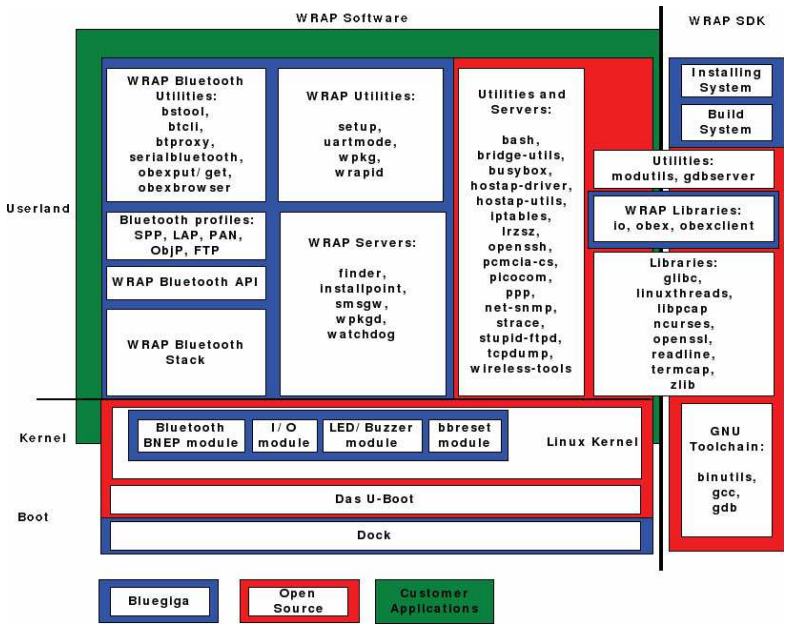

FIG. 2.1 – architecture système d'un point d'acces WRAP bluegiga

2.2 Architecture de communication pour WPAN

La plate-forme est articulée autour d'un serveur et de plusieurs points d'accès Bluetooth (cf Fig.2.2). En effet, il est évident qu'il n'est pas possible de couvrir complétement une station de métro en utilisant un seul point d'accès. Il est donc nécessaire de créer des cellules physiquement distinctes chacunes pilotées par un point d'accès. Ces cellules sont alors interconnectées par le biais d'un réseau IP et connectés vers un serveur centralisé.

Chacun de ces points d'accès permet alors d'assurer la communication entre les équipements Bluetooth mobiles pouvant se trouver sur deux réseaux différents (*i.e.* deux réseaux bluetooth distincts).

- Le serveur a pour rôle de stocker les messages provenant des bornes, d'authentifier des équipements mobiles du réseau et d'acheminer des messages à travers le réseau.
- Les points d'accès permettent d'identifier les nouvelles entités mobiles Bluetooth du réseau. Ils reçoivent les messages provenant des équipements mobiles et également les messages provenant d'autres entités : serveur, bornes, équipements mobiles. Ces messages sont ensuite acheminés vers les équipements mobiles du réseau.
- Les équipements mobiles (*e.g.* téléphones mobiles) possédent une fonctionnalités Bluetooth. Une application a été développée permettant l'envoi et la réception de messages vers les bornes.

Chaque point d'accès gère un WPAN. Chaque WPAN est physiquement disjoint des autres WPAN. Dans sa version simplifié le processus de communication suit le scénario suivant :

Lorsqu'une personne souhaite envoyer un message à un autre personne situé sur un autre WPAN, il utilise l'application embarquée. Le message est ensuite envoyé au point d'accès le plus proche, puis il est acheminé vers le serveur qui détermine la localisation du destinataire dans le réseau (*i.e.* sur quel point d'accès l'équipement mobile est connecté). Le message est alors transmis au point d'accès qui transmet à son tour à l'équipement mobile.

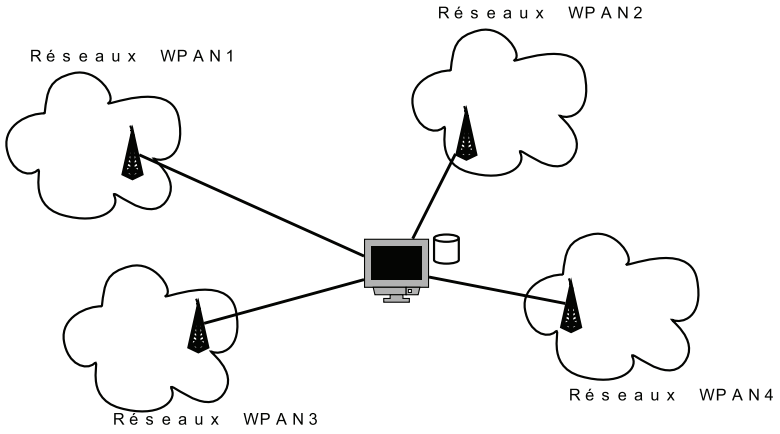

FIG. 2.2 – plate-forme de communications pour WPAN

Cette plate-forme a été testée pendant plusieurs mois dans une station du métro parisien. Nous avons remarqué un volume de données, qui est relativement faible et qui correspond au volume de données généré par une application de surveillance médicale en milieu hospitalier ou une application d'aide aux personnes. On peut également citer, l'environnement physique, composé dans le cas d'un environnement hospitalier ou d'une station de métro, de couloirs sur plusieurs niveaux. Enfin, l'aspect humain est également présent également certaines similitudes : mobilité, densité.

Cette plate-forme a permis de mettre en évidence certaines faiblesses des technologies sans fils WPAN dans des milieux clos.

La fiabilité des communications est notamment affectée. En effet, dans ce type de milieu, les ondes électromagnétiques sont fortement perturbées par les rebonds possibles sur les parois des couloirs. Egalement, la présence de multiples sources d'ondes électromagnétiques (*e.g.* éclairages, câblages électriques, équipements de radio mobile) affectent la propagation des ondes.

Il est donc nécessaire de mettre en place des mécanismes permettant de fiabiliser ces communications. C'est dans ce sens que se situent les travaux présentés dans la suite de ce document.

2.3 Etude de dimensionnement de la plate-forme

Une fois la plate-forme opérationnelle, il nous paraissait nécessaire d'en évaluer ses performances et de réaliser son dimensionnement. Cette étude théorique nous permet de valider les choix technologiques et architecturaux que nous avons choisi et implémenté. Cette étude va également nous permettre de définir les limites de la plate-forme et de spécifier de possibles règles de déploiement. Les faiblesses de l'architecture et des technologies mises en jeux pourront aussi être plus facilement identifiées. Nous devons

tout d'abord modéliser notre plate-forme. Celle-ci est composée de 4 points d'accès Bluetooth et d'un serveur. Les ressources de communications et de traitements entre les points d'accès et le serveur sont au regard de la technologie Bluetooth illimitées. Nous modéliserons donc uniquement le comportement des points d'accès.

2.3.1 Définition

Un point d'accès comporte une unité de traitement des données et 3 antennes Bluetooth. Chacune de ces antennes peut gérer 7 connexions Bluetooth simultanément. Dans nos scénarios, les équipements mobiles sont d'abord détectés par le point d'accès, puis ce dernier émet une requête vers l'équipement Bluetooth détécté. En retour, la personne répond à cette réquête. La personne peut alors choisir de rester dans la zone de couverture du point d'accès et continuer à émettre et recevoir des données.
Les points d'accès peuvent être modélisés comme des réseaux de files d'attente

$M/M/7/K_1$, $M/M/7/K_2$ et $M/M/7/K_3$ (cf Fig.2.3). La file d'attente K_1 modélise ici la taille de la zone de couverture d'un point d'accès. La valeur de K_1 est donc proportionnelle à la zone de couverture. Considèrant qu'une fois détectés, les personnes peuvent rester indéfiniment dans la zone de couverture, les files d'attente K_2 et K_3 sont considérées alors comme infinies.

On considère que les arrivées des personnes suivent une loi poissonienne de taux λ. Pour que notre étude soit plus réaliste, ce taux ne doit pas être fixe. En effet, au cours d'une journée, les arrivées de personnes ne sont pas constantes. Il y a plus de personnes le matin et le soir que pendant le courant de la journée. Par contre, au sein de ces différentes périodes, les taux d'arrivées des personnes suivent une loi poissonienne. Nous utiliserons donc différentes valeurs de λ afin de modéliser les différents instants de la journée. Nous avons considéré que la première file d'attente correspondait au temps passé entre l'arrivée dans la zone de couverture de l'antenne et la détection par cette antenne.

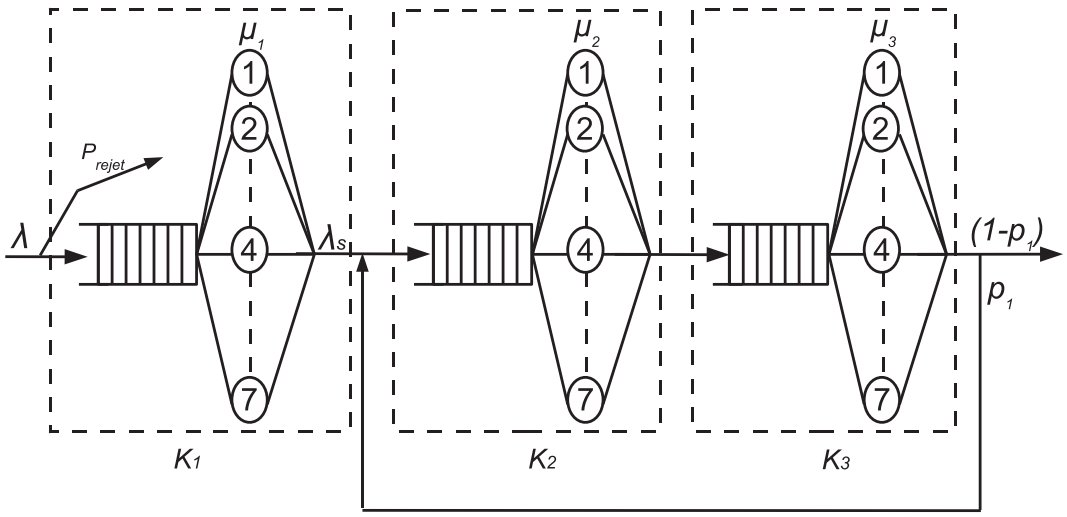

FIG. 2.3 – Représentation du point d'accès

Chacune des trois antennes pouvant réaliser 7 connexions simultanément, nous modélisons les 7 premières connexions par 7 serveurs. Les temps de service μ_1 des serveurs du système K_1 suivent tous une loi exponentielle. La détection d'un équipement Bluetooth n'est pas de durée fixe. Nos observations ont montré des temps de détections variant de 1 seconde à 7 secondes avec une valeur moyenne de 3 secondes. La détection n'étant pas immédiate, une personne peut franchir la zone sans pour autant être détecté par le point d'accès. La taille de la file d'attente dépend donc de la taille de la zone de couverture de l'antenne ; si celle-ci est petite, la taille de la file d'attente sera alors

réduite. Le nombre de clients dans ce premier système est K_1 et la taille de la file d'attente est $(K_1 - 7)$.

La probabilité P_{rejet} correspond donc à la probabilité qu'une personne franchisse la zone sans être détecté. Le trafic de sortie du système est fonction de la probabilité de rejet P_{rejet}. Elle est égale à $\lambda_s = \lambda(1 - P_{rejet})$.

Une fois qu'une personne est détectée, elle doit recevoir un message depuis le point d'accès. Nous considérons qu'une personne détectée reste dans la zone de couverture. Il n'y a plus de rejet et la file d'attente est considérée comme infinie. Nous avons donc le système K_2 ou K_2 représente le nombre de personnes dans le système. Une fois ce message reçu, elle doit à son tour envoyer un message vers le point d'accès. Le système K_3 est chargé de réceptionner ce message. Une fois celui-ci reçu, la personne peut choisir entre sortir du système ou rester à l'intérieur de celui-ci et attendre la réception d'un nouveau message. Il retournera alors dans la file d'attente du ystème K_2. La sortie de la personne est conditionnée par la probabilité p_1.

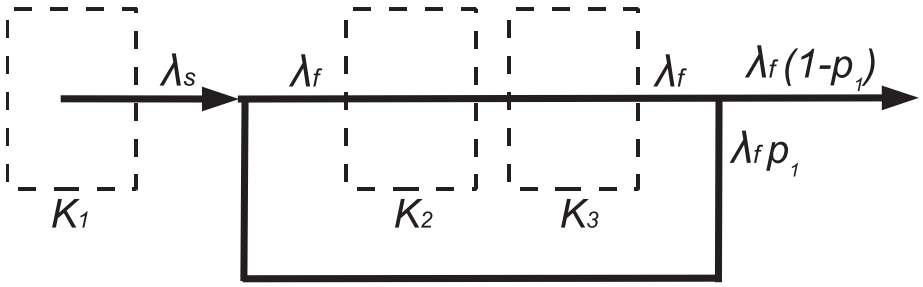

FIG. 2.4 – Calcul du processus d'arrivée λ_f

Comme dans le système K_1, les systèmes K_2 et K_3 possèdent chacuns 7 connexions pouvant réaliser respectivement 7 émissions et 7 réceptions en parallèle. Nous modéliserons ceci par 7 serveurs pour chaque système. Les données à émettre et les données reçues dépendent du profil de l'utilisateur et de ses requêtes. Les temps de service des systèmes K_2 et K_3 ne sont donc pas fixes. Ils suivent une loi exponentielle de taux μ_2 et μ_3. Le processus d'arrivée des clients dans les deux systèmes est λ_f.

Les personnes sortant du système K_3 peuvent revenir dans le système avant le système K_2. Le processus d'arrivée des clients dans ce système λ_f est donc différent du processus de sortie du système K_1 (cf Fig.2.4). λ_s est le débit du système en sortie de K_1. Il correspond au débit d'entrée dans le système K_1 auquel on retire les personnes rejetés si la file d'attente est pleine. Le débit de sortie est alors égal à $\lambda_s = \lambda \cdot (1 - P_{rejet})$. On rappelle que l'on considère les files d'attente de K_2 et K_3 comme étant infinies. Le processus d'arrivée des clients (*i.e.* λ_f) dans le système K_2 est donnée par l'équation Eqn.2.1.

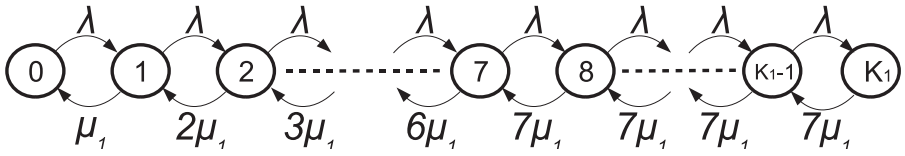

FIG. 2.5 – Chaîne de Markov associée à la file K_1

$$\begin{aligned} \lambda_f &= \lambda_s + p_1 \cdot \lambda_f \\ \lambda_f &= \lambda \cdot (1 - P_{rejet}) + p_1 \cdot \lambda_f \\ \lambda_f &= \frac{(1 - P_{rejet})}{1 - p_1} \cdot \lambda \end{aligned} \tag{2.1}$$

2.3.2 Chaîne de Markov associée

Nous avons défini notre file d'attente et nous devons maintenant définir la chaine de markov associée afin de réaliser l'étude de performances. La figure Fig.2.5 donne la représentation du système K_1. Les états de la chaine modélisent le nombre de personnes dans le système (file d'attente et serveurs).

Dès que l'on a au moins 8 personnes dans le système, il y a au moins une personne dans la file d'attente. A partir de 8 personnes, la transition pour revenir dans l'état précédent est donc $7\mu_1$. Pour avoir une personne de moins dans le système, il faut qu'une personne ait été détectée et sorte d'un des 7 serveurs (temps de service μ_1).

Nous utiliserons également cette représentation pour les systèmes K_2 et K_3. Dans ces derniers cas, λ sera alors remplacé par λ_f et les temps de service deviendront μ_2 et μ_3

2.3.3 Etude de performances

Nous allons tout d'abord déterminer la probabilité de rejet P_{rejet}. Celle-ci est égale à la probabilité d'avoir K_1 personnes dans le système K_1. On note $P(x)$ la probabilité stationnaire d'être dans l'état x (*i.e.* avoir x personnes dans le système). Ces probabilités sont calculées à partir des équations d'équilibre du système (*cf.* Eqn.2.2).

$$\begin{aligned}
P(1) &= \frac{\lambda}{\mu_1} \cdot P(0) \\
P(2) &= \frac{\lambda^2}{2! \cdot \mu_1^2} \cdot P(0) \\
P(3) &= \frac{\lambda^3}{3! \cdot \mu_1^3} \cdot P(0) \\
&\vdots \\
&\vdots \\
P(7) &= \frac{\lambda^7}{7! \cdot \mu_1^7} \cdot P(0) \\
P(n) &= \frac{\lambda^n}{n! \cdot \mu_1^n} \cdot P(0) \; pour \; 0 \leq n < 8
\end{aligned} \tag{2.2}$$

Lorsque l'on a plus de 7 personnes ($n \geq 8$) dans le système, l'équation d'équilibre est alors la même pour tous les autres états (*cf.* Fig.2.5)

$$\begin{aligned}
\frac{P(n+1)}{P(n)} &= \frac{\lambda}{7 \cdot \mu_1} \\
P(n) &= \left(\frac{\lambda}{7 \cdot \mu_1} \right)^{(n-8)} \cdot P(8) \\
P(n) &= \left(\frac{\lambda}{7 \cdot \mu_1} \right)^{(n-8)} \cdot \frac{\lambda^8}{7 \cdot 7! \cdot \mu_1^8} \cdot P(0) \\
P(n) &= \frac{\lambda^n}{7! \cdot 7^{n-7} \cdot \mu_1^n} \cdot P(0)
\end{aligned} \tag{2.3}$$

On obtient donc le système suivant :

$$P(n) = \left\{ \begin{array}{lll}
P(n) &= \dfrac{\lambda^n}{n! \cdot \mu_1^n} \cdot P(0) \; pour \; 0 \leq n < 8 \\
P(n) &= \dfrac{\lambda^n}{7! \cdot 7^{n-7} \cdot \mu_1^n} \cdot P(0) \; pour \; n \geq 8
\end{array} \right)$$

En utilisant les équations d'équilibre et la condition de normalisation $\sum_{n=0}^{K_1} P(n) = 1$, on en déduit $P(0)$:

$$P(0) = \frac{1}{\sum_{n=1}^{7} \frac{\lambda^n}{n! \cdot \mu_1^n} + \sum_{n=8}^{K_1} \frac{\lambda^n}{7! \cdot 7^{n-7} \cdot \mu_1^n}} \tag{2.4}$$

La probabilité de rejet P_{rejet} est équivalente à la probabilité d'avoir la file d'attente pleine et tous les serveurs occupés (*i.e.* $P(K_1)$). On calcul donc $P(K_1)$ avec $K_1 \geq 8$.

$$P_{rejet} = P(K_1) = \frac{\lambda^{K_1}}{\left(7! \cdot 7^{K_1-7} \cdot \mu_1^{K_1}\right) \cdot \left(\sum_{n=1}^{7} \frac{\lambda^n}{n! \cdot \mu_1^n} + \sum_{n=8}^{K_1} \frac{\lambda^n}{7! \cdot 7^{n-7} \cdot \mu_1^n}\right)} \tag{2.5}$$

$$P_{rejet} = P(K_1) = \frac{\rho_1^{K_1}}{(7! \cdot 7^{K_1-7}) \cdot \left(\sum_{n=1}^{7} \frac{\rho_1^n}{n!} + \sum_{n=8}^{K_1} \frac{\rho_1^n}{7! \cdot 7^{n-7}}\right)} \text{ ou } \rho_1 = \frac{\lambda}{\mu_1} \tag{2.6}$$

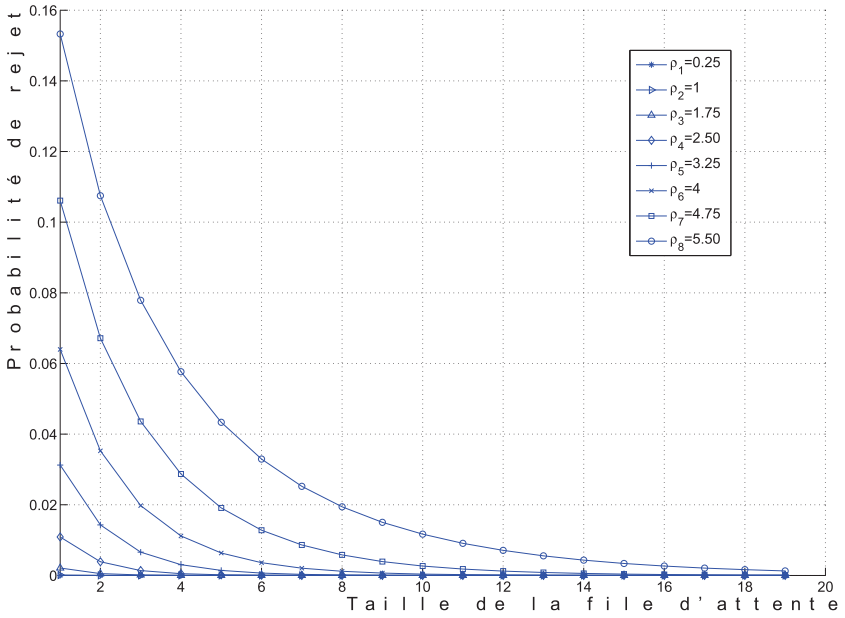

FIG. 2.6 – Probabilité de rejet en fonction de la taille de la file d'attente

La figure Fig.2.6 représente la probabilité de rejet P_{rejet} en fonction de la taille de la file d'attente et en fonction de ρ. Tout au long d'une journée, les taux d'arrivées des personnes varient tous en restant relativement stables lors de certaines périodes : matin, soir, milieux de journée etc ... Nous avons calculé les probabilités de pertes en fonction de différentes valeur de λ et par conséquent différentes valeurs de ρ. ρ représente alors la charge d'un serveur.

Pour notre étude, nous avons modélisé la file d'attente comme étant le temps de séjour des personnes dans la zone de couverture de l'antenne. Ce temps de séjour est donc fonction de la taille de la zone de couverture et de la vitesse de déplacement moyenne d'une personne. On voit clairement que pour moins de 2 arrivées d'usagers par seconde, la probabilité de rejet reste très faible (¡1%) et cela quelque soit le temps de séjour.

Pour des taux d'arrivées plus importants qui pourraient correspondre aux heures de pointes par exemple, la probabilité de rejet est un peu différentes. Si les personnes ne reste dans la zone de couverture que pendant 2 secondes, les probabilités d'avoir des clients non détectés peuvent être de l'ordre de 10%. A partir de 10 secondes, les probabilités d'avoir des personnes non détectés peuvent être inférieur à 2%. Dans le déploiement que nous avons effectué, les zones de couvertures ont été évaluées à plus de 10 secondes. On a donc une probabilité de rejet inférieure à 1 % quelque soit le taux d'arrivées.

Nous nous intéressons maintenant au temps moyen de détection d'une personne (temps dans la file d'attente compris). Pour cela, on utilise la loi de Little [58]. Celle-ci nous donne la relation entre le nombre moyen de client dans le système Q, le temps moyen total du séjour R et le débit moyen du système X. On a donc $Q = R \cdot X$.

La file d'attente du système ayant une taille limitée et des pertes pouvant subvenir, le débit X est donc différent de λ et est égal à λ_s.

Le temps moyen de séjour d'une personne (R) dans un système est la somme du temps moyen de séjour dans la file d'attente (R_w) et d'un temps moyen de service d'un serveur ($1/\mu_1$). Il faut donc, dans un premier temps, déterminer le nombre moyen de client dans la file d'attente (Q_w). On rappelle que λ_s correspond au débit effectif d'arrivée des clients dans la file d'attente. λ_s est alors égale à $\lambda(1 - P_{rejet})$.

$$
\begin{aligned}
Q_w &= \sum_{n=7}^{\infty} (n-7) \cdot P(n) \\
Q_w &= \sum_{n=7}^{\infty} (n-7) \cdot \frac{\lambda_s^n}{7! \cdot 7^{n-7} \cdot \mu_1^n} \cdot P(0) \\
Q_w &= \frac{\lambda_s^8}{7! \cdot 7 \cdot \mu_1^8} \sum_{n=7}^{\infty} (n-7) \cdot \left(\frac{\lambda_s}{7 \cdot \mu_1} \right)^{n-8} \cdot P(0) \\
Q_w &= \frac{\lambda_s^8}{7! \cdot 7 \cdot \mu_1^8} \cdot \frac{1}{\left(1 - \frac{\lambda_s}{7\mu_1}\right)^2} \cdot P(0) \\
Q_w &= \frac{\lambda_s^8}{\mu_1^8 \cdot 6! \cdot (7 - \frac{\lambda_s}{\mu_1})^2} \cdot P(0)
\end{aligned} \tag{2.7}
$$

Le temps moyen de séjour dans la file d'attente est donc

$$
R_w = \frac{\lambda_s^7}{\mu_1^8 \cdot 6! \cdot \left(7 - \frac{\lambda_s}{\mu_1}\right)^2} \cdot P(0) \tag{2.8}
$$

Et le temps moyen de séjour dans le système K_1

$$
R = \frac{\lambda_s^7}{\mu_1^8 \cdot 6! \cdot \left(7 - \frac{\lambda_s}{\mu_1}\right)^2} \cdot P(0) + \frac{1}{\mu_1} \tag{2.9}
$$

La figure Fig.2.7 représente le temps moyen de séjour en fonction de λ et de la taille de la file d'attente. Comme pour le calcul de la probabilité de rejet, nous avons fait varier la fréquence d'arrivée des usagers en proximité du point d'accès. On voit clairement que le temps moyen de séjour est relativement limité (inférieur à 4 secondes) lorsque les arrivées d'usagers sont inférieurs à 2 par seconde. Dans le pire des cas, le temps d'attente moyen calculé n'excède pas 6 secondes. Il est a noté que les temps calculé prennent en compte la probabilité de rejet. C'est à dire qu'il n'y a plus de rejet de client à ce niveau là. Les clients ayant traversés la zone de couverture dans être détecté par le point d'accès n'apparaissent pas dans ce calcul. La taille de la file d'attente détermine ici le temps d'attente et non plus la probabilité que l'usager soit "rejeté".

On s'intéresse maintenant aux systèmes K_2 et K_3. On rappelle que ces systèmes prennent en compte les usagers qui ont été detectés (*i.e.* dans le système K1). Le système K_2 modélise l'envoie d'une requête vers les équipements mobiles et le système K_3, la réception de la réponse à cette requête (du mobile vers le point d'accès). Le nombre moyen de client dans ces systèmes ainsi que le temps moyen de séjour sont des

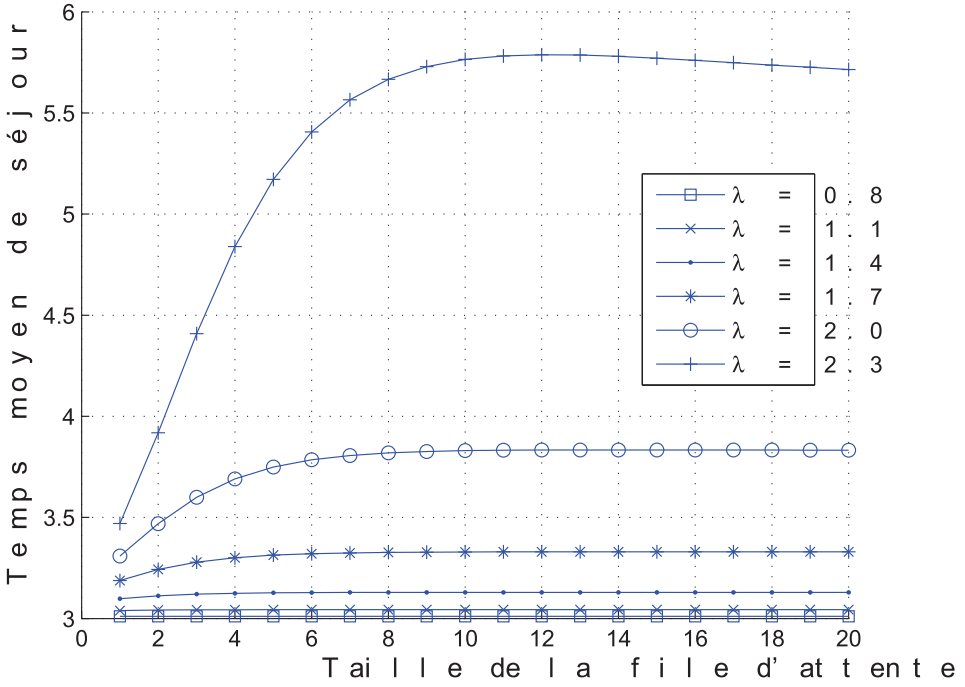

FIG. 2.7 – Temps moyen de séjour dans le système K1

données importantes et permettent d'évaluer notre solution et de spécifier ses limites de fonctionnement. Elles nous permettent notamment d'évaluer les temps d'émissions et de réceptions des requêtes et donc les temps d'attentes des usagers. Pour obtenir ces évaluations, on utilise une fois de plus la loi de Little.

Donc notre cas, les systèmes K_2 et K_3 sont considérés comme étant à file infini. Le débit en entrée des systèmes est alors le même qu'en sortie. On a donc $X = \lambda_f$

On applique la loi de Little à la file d'attente. On obtient donc

$$R_w = \frac{Q_w}{\lambda_f}$$

On doit déterminer le nombre moyen de client dans la file d'attente (Q_w). On applique alors la méthode utilisé précedemment pour le système K_1.

$$Q_w = \frac{\lambda_f^8}{\mu_x^8 \cdot 6! \cdot (7 - \frac{\lambda_f}{\mu_x})^2} \cdot P(0) \tag{2.10}$$

On peut alors déterminer le temps moyen de séjour dans la file d'attente (R_w) et le temps moyen de séjour dans le système (R).

$$R_w = \frac{\lambda_f^7}{\mu_x^8 \cdot 6! \cdot \left(7 - \frac{\lambda_f}{\mu_x}\right)^2} \cdot P(0) \tag{2.11}$$

$$R = \frac{\lambda_f^7}{\mu_x^8 \cdot 6! \cdot \left(7 - \frac{\lambda_f}{\mu_x}\right)^2} \cdot P(0) + \frac{1}{\mu_x} \tag{2.12}$$

Le nombre de moyen de clients dans le système (Q) est donc :

$$Q = R \cdot X$$

$$Q = \left(\frac{\lambda_f^7}{\mu_x^8 \cdot 6! \cdot \left(7 - \frac{\lambda_f}{\mu_x}\right)^2} \cdot P(0) + \frac{1}{\mu_x} \right) \cdot \lambda_f \tag{2.13}$$

Le nombre moyen de clients dans le système est donc fonction de λ_f, μ_x et $P(0)$. Pour le calcul de P(0), on applique la méthode utilisée pour le système K_1 en prenant en compte le fait que les files d'attentes des systèmes K_2 et K_3 sont toutes deux infinies. La durée moyenne de service μ_x est ici $\mu_2 = 4$ pour le système K_2 et $\mu_3 = 3$ pour le système K_3. On obtient alors, pour les systèmes K_2 et K_3, la valeur P(0) donnée dans l'équation Eqn2.14.

$$P(0) = \frac{1}{\sum_{n=0}^{6} \frac{\lambda_f^n}{n! \cdot \mu_x^n} + \frac{\lambda_f^7}{6! \cdot \left(7 - \frac{\lambda_f}{\mu_\times}\right) \cdot \mu_x^7}} \tag{2.14}$$

Ceci nous permet alors de calculer le temps de sejour moyen et le nombre moyen de client dans les systèmes K_2 et K_3.

La figure 2.8 montre le nombre moyen de personnes dans le système K_2. Dans les calculs que nous avons effectués nous avons positionné la probabilité de retour dans le système (p_1) à 0,5. C'est à dire que lors de l'envoi d'un message, la personne attendra une réponse du point d'accès dans 50% des cas. Sinon, elle sortira du système. La figure 2.9 nous donne le temps moyen de séjour dans le système K_3. C'est à dire, le temps d'attente de réception d'un message. C'est une information pertinante car elle permet de juger de la réactivité de la plate-forme. Un temps de séjour trop long pourrait avoir comme conséquence le départ des personnes avant de recevoir une réponse du point d'accès. Dans notre étude, on observe un temps moyen de séjour inférieur à 6 secondes, même lors d'une forte affluence.

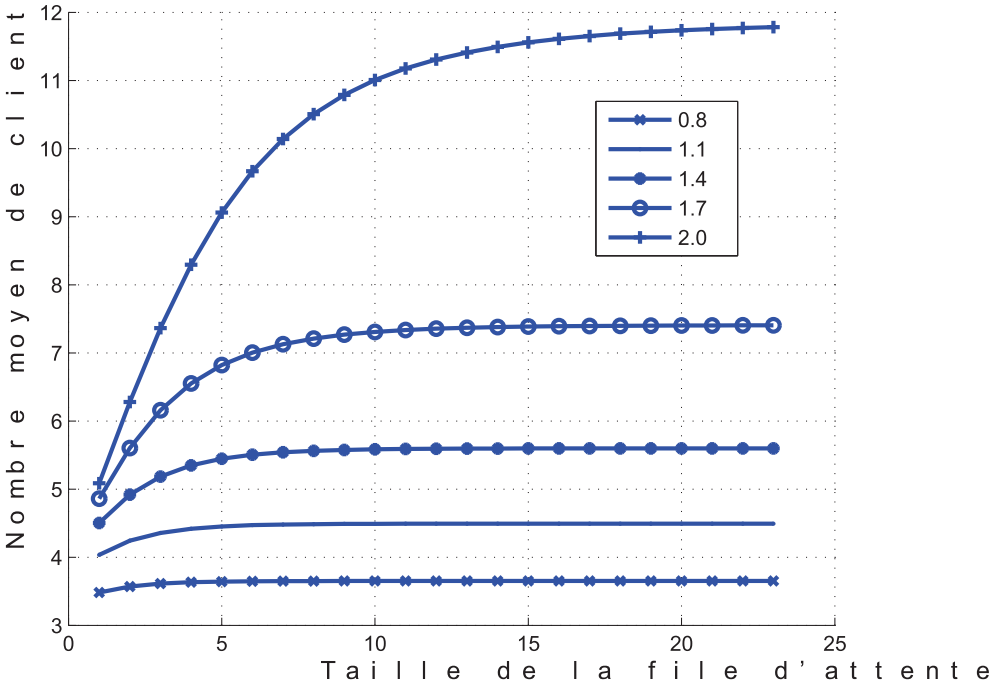

FIG. 2.8 – Nombre moyen de personne dans le système K2

2.3.4 Conclusion

L'étude de dimensionnement que nous venons de réalisé nous a permis de valider les différents aspects de notre plate-forme : la technologie, l'architecture et l'intégration en elle-même.

Cette étude nous fournit des informations pertinentes sur l'implantation de nouveaux points d'accès. Elle permet notamment de définir des zones de couvertures limites qui permettent de planifier les futures implantations en fonction de l'environnement. Cette évaluation nous à notamment permis d'estimer le nombre moyen de personne se trouvant à proximité de la borne et donc de d'éviter certains lieux d'implanation des points d'accès. En effet, si le nombre de moyen de personnes est elevé, il est alors nécessaire de ne pas installer de point d'accès dans une zone de fort passage, au risque de perturber le flux de personne.

Cette étude nous a également permit de mettre en avant une faiblesse de la technologie en elle-même quand à la détection d'équipement Bluetooth. Dans certains tests que nous avons réalisé, il est apparu que certains équipements Bluetooth avait franchi la zone de couverture sans pour autant être détectés. Ces cas de non-détection sont très

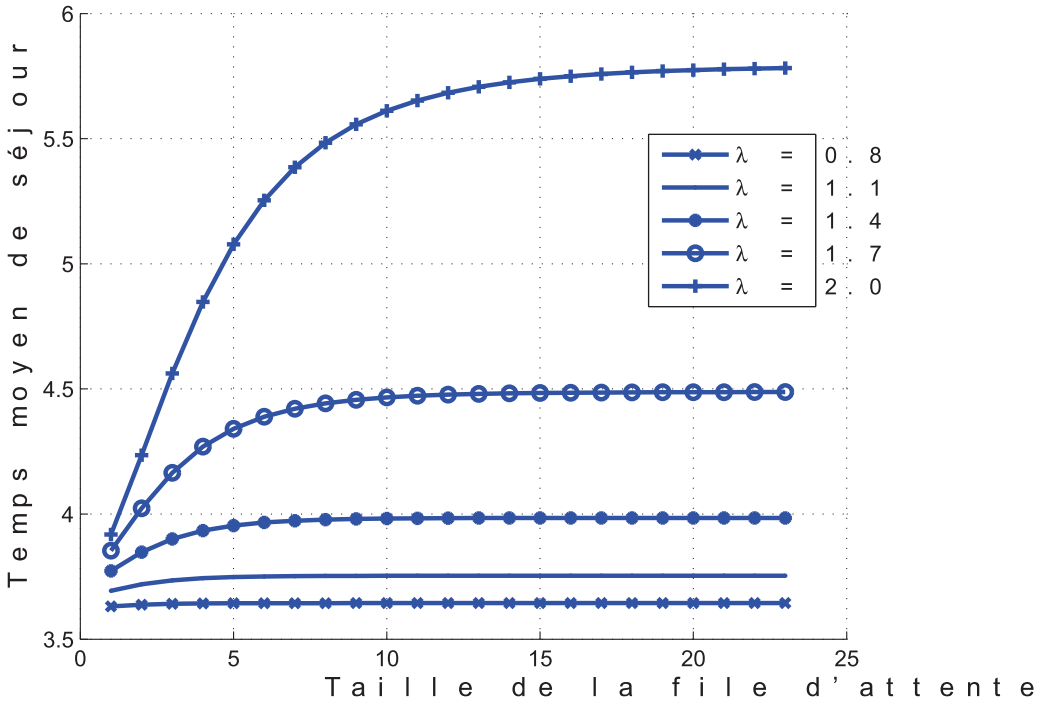

FIG. 2.9 – Temps moyen de séjour dans le système K3

certainement le fait du processus de détection Bluetooth. L'étude réalisé fait abstraction de ces quelques cas car ils sont très anecdotiques et ne réprésente pas l'ensemble des tests que nous avons réalisé.

Une accélération du processus de détection par l'équipement Bluetooth pourrait permettre de réduire la probabilité de perte de notre solution. Cette modification, nécessitant la modification de la pile protocolaire bluetooth, n'a put être réalisé dans les temps impartis. Ceci pourrait permettre d'implanter des équipements dans des zones de couvertures très réduites.

Chapitre 3

Mécanisme de réduction de l'utilisation de la bande passante

La surveillance des signes vitaux des patients dans un hôpital est une partie importante du traitement. Les réseaux de capteurs sans fils sont considérés comme une solution potentielle dans le futur permettant de réduite fortement la charge de travail du personnel médical. Comme présenté dans la section précédente, l'utilisation des technologies sans fils actuelles nécessite des adaptations, les applications médicales requérant des contraintes spécifiques à ce domaine. Les problèmes de bande passante, de consommation d'énergie et de fiabilité des communications sont des thématiques qui nécessitent d'adapter les solutions existantes.

Dans cette section, nous présenterons une solution basée sur une compression des données et une aggregation de ces données permettant de réduire la consommation d'énergie dans les réseaux de capteurs [35]. Cette solution permet aussi de réduire l'utilisation du canal de communication et donc d'augmenter la bande passante disponible. La fiabilité des communications est également optimisée.

3.1 Architecture d'un capteur

Un réseau de capteurs est, par définition, constitué de capteurs. Précédemment, nous nous étions intéressés aux réseaux en général. Dans cette section, nous nous interesserons au capteur en lui-même. Il existe, sur le marché actuel, de nombreuses plateformes de capteurs : Micaz [107], TelosB, Moteiv [68],Eyes [31], iMote [45], WSN430 [106].

L'architecture de ces plate-formes est relativement simple et assez semblable. Les capteurs sont, pour la plupart, constitués d'un microprocesseur, d'un chipset radio, d'un espace de stockage et d'un appareil de mesure. Dans notre laboratoire nous possédons notamment des capteurs Micaz. Leur architecture est donnée en figure Fig.3.1.

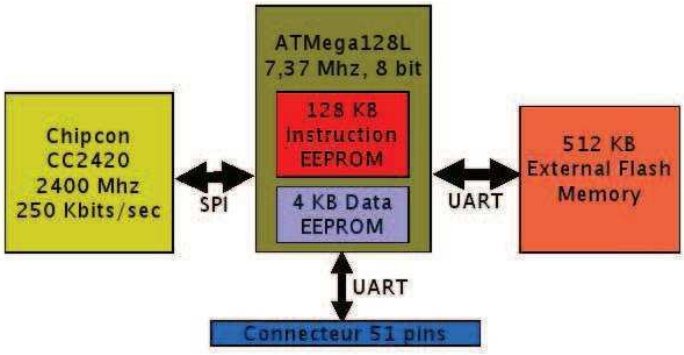

FIG. 3.1 – Architecture d'un capteur Micaz

Ces équipements possèdent des ressources très limitées. Dans le cas du Micaz, la fréquence de fonctionnement du microprocesseur Atmel128 n'est que de 7,3 MHz et son espace de stockage est de l'ordre de 500 Ko. Son chipset radio CC2420, compatible IEEE 802.15.4, permet d'atteindre un débit théorique de 250 Kbits/s et fonctionne dans la bande des 2.4 GHz. Cette "limitation" permet cependant d'obtenir des équipements consommant peu d'énergie. Cette consommation totale d'enérgie se partage majoritairement entre deux entités du capteur : le chipset radio et le microprocesseur. Pour réduire la consommation d'énergie, une solution possible serait d'optimiser le comportement des capteurs afin de réduire la consommation de l'un de ces deux composants.

3.2 Algorithme de décision

Dans le but de surveiller les constantes d'un patient, chaque capteur effectue une mesure (ECG, pression sanguine, température, etc..) de façon périodique. Cette mesure doit ensuite être envoyée à l'équipe médicale afin qu'elle puisse surveiller les constantes médicales et décider ou non d'intervenir.

Dans certains cas, les mesures effectuées peuvent varier faiblement entre deux instants rapprochés. Le noeud doit pouvoir prendre la décision de ne pas envoyer cette donnée et ainsi de conserver l'énergie qui aurait été dépensé pour cette transmission. La donnée n'est pas envoyée immédiatement mais elle devrat tout de même parvenir à l'équipe médicale en temps voulu. Dans le cas d'une évolution significative (information critique) d'une constante, la transmission doit être immédiate.

Les données qui n'ont pas été émises immédiatement sont compressées en utilisant l'algorithme défini par Lempel *et al.* [112]. C'est un algorithme de compression qui est non-destructeur (*i.e.* il n'y a pas de pertes de données lors des phases de compression et décompression). Cette compression permettra alors de réduire le volume de données

émis, donc de réduire l'énergie utilisée pour la transmission, et de limiter l'utilisation de la bande passante [48]. La figure 3.2 montre que le taux de compression dépend de la variabilité des valeurs mesurées ainsi que de la taille de la fenêtre de compression. Cette fenêtre correspond au temps pendant lequel les données seront stockées. On peut remarquer que le taux de compression augmente lorsque la fenêtre de compression augmente. En effet, plus le volume de données à compresser est important, plus on augmente la probabilité d'avoir une redondance d'informations dans ces données. Il faut alors définir une fenêtre suffisante pour compresser suffisament de données et obtenir ainsi un taux de compression intéressant.

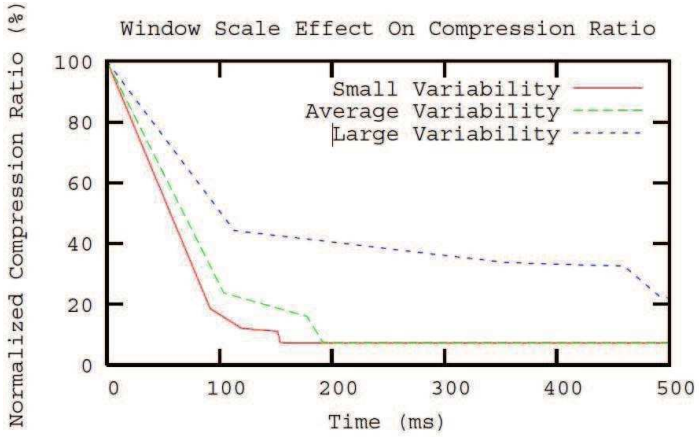

FIG. 3.2 – Taux de compression des données en fonction du temps disponible pour la compression et du cardinal des valeurs mesurées (Variabilité)

Il est donc nécessaire de stocker les données pendant un temps t. Ce temps de stockage est un paramètre qui est fonction de la taille des données et de l'espace de stockage disponible. Ce temps est également fonction de l'application. En effet, même si l'espace de stockage est suffisant, il est nécessaire d'envoyer les données au destinataire avant qu'elle ne soit périmée. Ainsi, chaque donnée possède une date limite, *deadline*, avant laquelle elle doit être impérativement envoyée. Une fois ce delai atteint, les données présentes en mémoire sont compressées et le paquet nouvellement crée est alors transmis à son destinataire. L'algorithme de cette application est presenté en figure 3.3.

3.3 Modèle analytique

Dans cette section, nous présenterons notre modèle de consommation d'énergie d'un noeud dans un réseau de capteurs. Supposons que le nombre de données à envoyer est n, que la i^{eme} donnée a une taille $D(i)$. Soit :

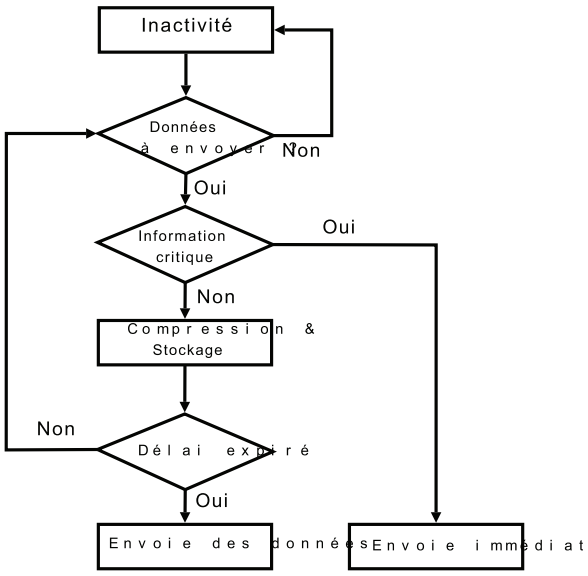

FIG. 3.3 – Algorithme de décision pour la compression de données

– C_c : l'énergie nécessaire pour compresser un octet de données
– C_t : l'énergie nécessaire pour transmettre un octet de données
– w : le taux de compression
Définissons les paramètres suivants :

- $x(i) = 1$ si la donnée n'est pas jugée critique par le capteur.
  $x(i) = 0$ sinon (la donnée devra être compressée).

La consommation d'énergie est partagée entre le mécanisme de compression des données et le mécanisme d'envoi de ces données. L'énergie consommée E par un noeud pour traiter une donnée de n octets est donnée par l'équation Eqn.3.1.

$$E = C_c \sum_{i=1}^{n} \Big[\big(1 - x(i)\big) D(i) \Big] + C_t \sum_{i=1}^{n} \Big[\big(1 - x(i)\big) w D(i) + x(i) D(i) \Big] \qquad (3.1)$$

Le terme $1 - x(i)$ assure que les données non prioritaires sont compressées.
Les conditions suivantes doivent être respectées.
– $\sum_{i=1}^{n} (1 - x(i)) \cdot w \cdot D(i) <$ taille du buffer
– Les données stockées ne peuvent être supérieures à la taille du buffer disponible.

La figure Fig.3.4 donne l'énergie nécessaire pour transmettre des données en fonction de certains paramètres (taux de compression w et criticabilité des données x). Elle a été obtenue à partir de l'équation Eqn.3.1 en la modélisant sous matlab.

Nous avons également modélisé l'énergie utilisée par le capteur lorsque l'algorithme de décision n'est pas utilisé.

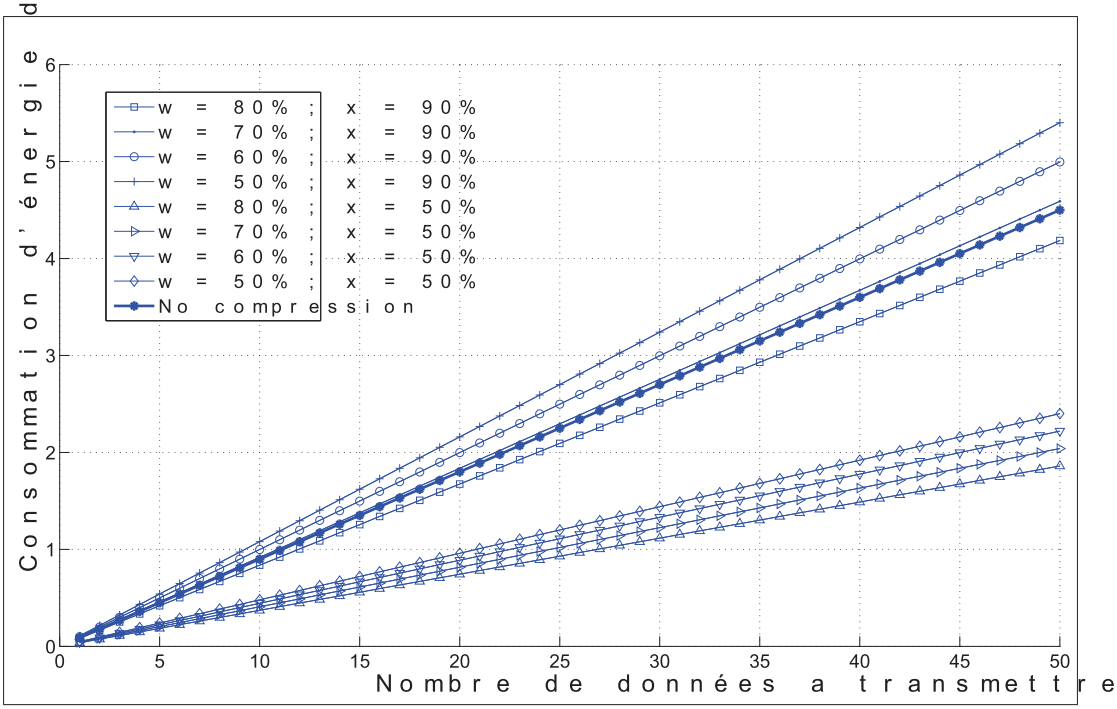

FIG. 3.4 – Consommation d'énergie en fonction du nombre de données à transmettre ; w :taux de compression, x : criticabilité des données

On voit clairement que la criticabilité des données est un paramètre primordiale. Si 90 % des données sont critiques et ne doivent pas être compressé, l'intérêt de l'utilisation de notre solution dépend alors de l'algorithme de compression. S'il est performant et permet d'atteindre des taux de compressions de 80 %, alors notre solution permet d'économiser de l'énergie. Si ce taux de compression est moindre, l'utilisation de notre algorithme n'est pas recommandée. Si les données générées ne sont critique que pour la moitié d'entre elles, notre solution est plus économiques qu'un envoi sans compression. Et cela, quelque soit le taux de compression atteint par l'algorithme.

Nous devons donc utiliser un algorithme de compression relativement puissant. Nous devons également limiter la criticabilité des données afin de ne pas avoir à transmettre trop de données non compressées.

3.4 Evaluation

L'évaluation de notre algorithme de décision a été réalisée à l'aide du système d'exploitation TinyOS [93] et du simulateur Avrora [11].

3.4.1 TinyOS

TinyOS est un système d'exploitation développé par le département WEBS [104] de l'Université de Californie à Berkeley. Ce système d'exploitation, très répandu parmi la communauté *capteurs*, fonctionne sur les nombreuses plate-formes existante.

C'est un système d'exploitation événementiel basé sur un modèle concurrentiel d'éxecution : les applications définies sous TinyOS sont constituées de différents composants qui sont exécutés par le microprocesseur selon des règles définies par un ordonnanceur. Ce dernier va alors ordonnancer les différentes tâches et événements issus de ces composants.

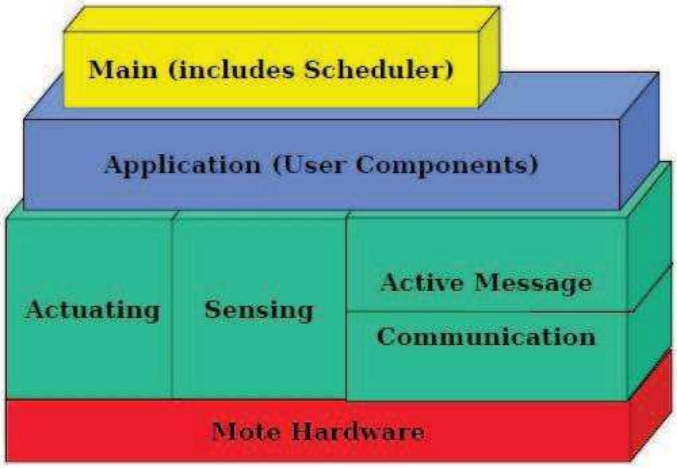

FIG. 3.5 – Architecture de TinyOS

Cet ordonnancement est relativement simple : il est basé sur une simple file d'attente FCFS (First Come First Serve). Celle-ci contient 7 places. Lorsqu'une tâche survient elle est placée dans la file d'attente afin d'être exécutée. La notion de préemption n'existe pas entre tâches. Une tâche ne peut pas interrompre une autre tâche qui serait en train d'être éxecutée et ne peut pas passer devant une autre tâche dans la file d'attente. Cependant, lorsqu'un événement survient, il est préemptif vis-à-vis des tâches présentes. Il va donc interrompre la tâche qui est actuellement exécutée par le microprocesseur et effectuer le traitement correspondant à l'événement survenu. Les événements ne sont pas, à l'instar des tâches, préemptifs entre eux. Ils ne peuvent pas

s'interrompre mutuellement. Une application tinyOS est écrite en utilisant deux types de composants : tâche et événement.

- tâche Les tâches sont des blocs d'instructions. Dans la philosophie de programmation de TinyOS, elles sont principalement chargées du traitement de l'information, elles réalisent des fonctions de calcul, de stockage d'informations, de lecture. Comme elles peuvent être interrompues par un événement, leur temps d'execution doit être le plus court possible. Les tâches longues sont proscrites en TinyOS. Il est alors recommandé de séparer une tâche en plusieurs sous-tâches. C'est la notion de *split-phase* : l'acquisition d'une donnée se déroule en deux temps. La première tâche consiste à faire une demande de lecture de donnée et la deuxième tâche à lire cette donnée lorsqu'elle est prête. Ainsi, on n'attend pas que la donnée soit disponible est ainsi risqué d'être interrompu par un événement.
- événement Les événements sont des composants proches du principe d'interruption que l'on trouve dans les systèmes informatiques classiques. Ces événement peuvent être déclenchés par des interruptions matérielles ou des interruptions logicielles (*e.g.* timer). Lorsqu'un événement se produit, le microprocesseur doit le plus rapidement possible exécuter le code correspondant à l'événement survenu.

Ces composants interagissent entre eux par le biais d'interfaces. La bibliothèques actuelle de TinyOS comporte plus d'une centaine de composants qui implémentent des fonctionnalités de transmission, de calcul, de lecture/ecriture, etc ... Le code développé sous TinyOS utilise donc ces bibliothèques et les lie ensemble par l'intermédiaire de composants. Ces dernier sont écrits en Nesc.

Le Nesc est un langage de programmation spécialement crée pour les réseaux de capteurs et TinyOS. Il est également issu de l'université de Berkeley. C'est un langage dont la syntaxe est proche du C, mais qui permet de satisfaire au modèle concurrentiel de TinyOS. Il est également plus restrictif pour satisfaire les contraintes matérielles des réseaux de capteurs. Les allocations dynamiques et les pointeurs notamment sont très fortement déconseillés car mal utilisés, ils peuvent rendrent l'application instable. Une application Nesc est donc composée d'un ensemble de composants liés entre eux par des interfaces bi-directionnelles.

Actuellement, TinyOS est toujours maintenu par l'université de Berkeley et est utilisé par plus de 500 laboratoires et centre de recherches dans le monde. Il existe actuellement deux versions de ce système d'exploitation : tinyos-1.x et tinyos-2.x.

3.4.2 Avrora

Avrora est un simulateur développé par le groupe Compiler de l'Université de Californie à Los Angeles. Ce simulateur permet d'évaluer avec précision le fonctionnement des applications développées pour les réseaux de capteurs. Il permet notamment de

connaître la consommation d'énergie en Joule de chaque composant du noeud : microprocesseur, radio, leds, espace de stockage ... Cela nous permet d'évaluer avec précision l'impact de l'algorithme de décision sur la consommation d'énergie de chaque composant du capteur.

Dans le cadre de l'analyse de notre solution, nous avons donc développé notre solution sous TinyOS. Nous avons basé notre travail autour de l'algorithme de compression de Lempel et Ziv : LZ77 [112].

3.4.3 LZ77

L'algorithme LZ77 est un algorithme de substitution classique. Il repose sur la constitution d'un dictionnaire des phrases rencontrées lors de la compression d'une donnée et du référencement de la position des autres occurences de cette même phrase dans la donnée. L'avantage de cet algorithme est, dans notre cas, qu'il ne nécessite pas de ressources importantes et que son implémentation est relativement réduite (moins de 50 lignes de codes).

3.5 Résultats

Afin d'évaluer notre algorithme, l'utilisation d'un émulateur [49] fournissant plusieurs constantes médicales (ElectroCardioGramme, actimètre, positionnement relatif) a été considérée. Cette émulateur a été développé par le département EPH de l'Institut Télécom Sud Paris. Cet émulateur permet d'obtenir des signaux "variables". Cela nous permet donc de modifier la variabilité des mesures et ainsi de voir son influence sur les performances de la compression et de l'algorithme de décision.

Nous nous intéresserons, dans un premier temps, à la consommation d'énergie et à l'influence de la variabilité sur les performances. Les figures Fig.3.6 et Fig.3.7 montrent la consommation d'énergie du microprocesseur et du chipset radio en fonction de la variabilité des mesures. La consommation engendrée par le microprocesseur lors de la compression est relativement peu influencée par la variabilité. En effet, lors de la compression des données, la variabilité des mesures ne modifie que peu les instructions exécutées par le microprocesseur. Une faible variabilié réduit logiquement le nombre d'instructions à éxécuter, mais cela n'est pas significatif. On peut considérer que la consommation d'énergie du microprocesseur n'est donc pas fonction de la variabilité des données.

De même pour la consommation d'énergie dans le chipset radio : la compression influe sur le volume de donnée à envoyer en le réduisant. Dans un premier temps, on vérifie si la variabilité des données influe sur l'énergie nécessaire à la transmission.Cependant, la différence de variabilité influe peu sur l'énergie utilisé pour une transmission.

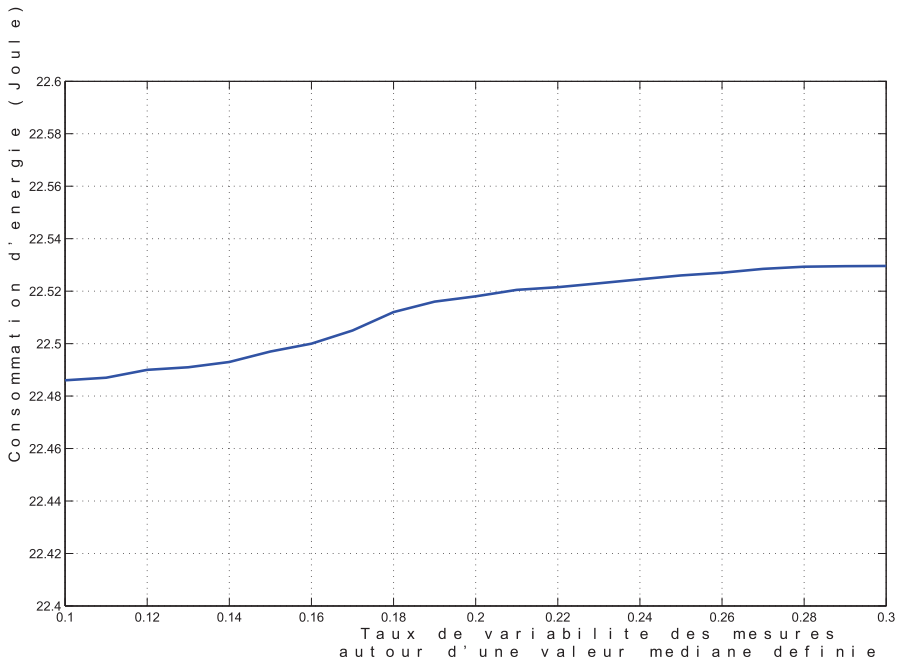

FIG. 3.6 – Consommation d'énergie dans le microprocesseur d'un capteur lors d'une compression en fonction de la variabilité des données en entrées

Les résultats ont été obtenus après 1000 secondes d'expérimentation et en générant un débit théorique de 4 Koctets/s. Ce débit permet de supporter la majorité des applications médicales [36].

Afin d'évaluer l'algorithme de décision et de quantifier ses performances vis à vis d'une solution classique, nous avons determiné deux scénarios. Dans le premier scénario, les mesures effectuées sont envoyées immédiatement, sans utiliser l'algorithme de décision. Dans le deuxième cas, l'algorithme de décision est utilisé. Le tableau 3.1 résume la consommation d'énergie d'un noeud pendant 1000 secondes.

Dans le premier cas, la consommation d'énergie est répartie équitablement entre le microprocesseur et le chipset radio. Le microprocesseur effectue quelques tâches simples : lecture/écriture, création de packet, etc ... Dans le cas de l'utilisation de l'algorithme de décision, le microprocesseur effectue des opérations pour réaliser la compression des données. Cela a pour corollaire une augmentation de sa consommation d'énergie. Néanmoins, les données sont maintenant compressées, leur volume est moindre et le chipset radio a donc moins d'information à envoyer. L'énergie nécessaire pour l'envoi des données est alors réduite et la consommation d'énergie du chipset radio également.

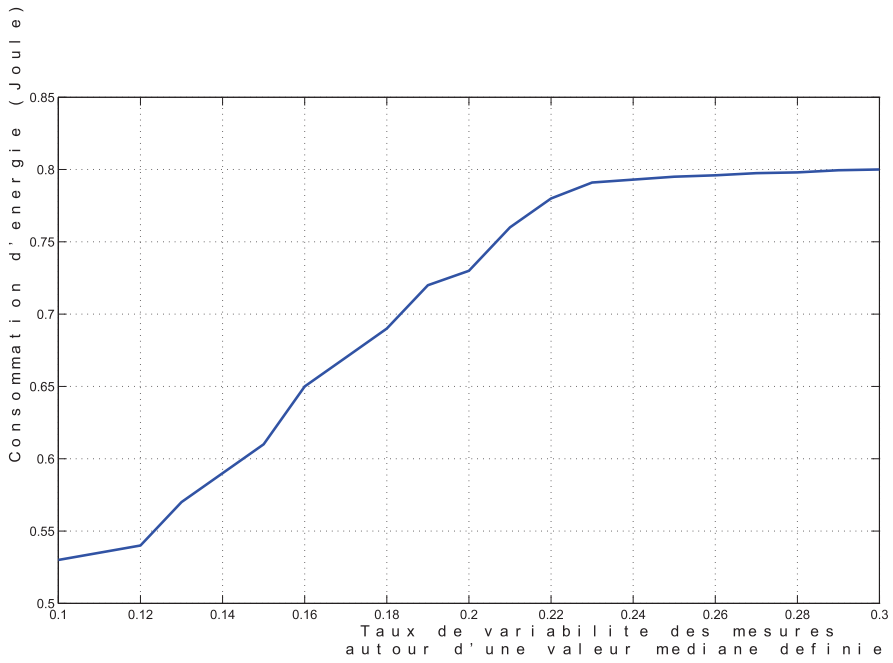

FIG. 3.7 – Consommation d'énergie dans le chipset radio d'un capteur lors de l'utilisation de la compression en fonction de la variabilité des données en entrée

	Consommation d'énergie CPU	**Consommation d'énergie transmission**	**Consommation Totale**
Sans compression	11.513 J	13.055 J	24.568 J
Avec compression	22.664 J	0.229 J	22.893 J

TAB. 3.1 – Consommation d'énergie en Joule d'un capteur pendant 1000 secondes avec et sans algorithme de décision

	Octets de données envoyés
Sans compression	6.800.000
Avec compression	650.000

TAB. 3.2 – Consommation de la bande passante pour un capteur pendant 1000 secondes avec et sans algorithme de décision

Finalement, la consommation totale d'énergie dans le capteur est réduite dans le cas de l'utilisation de l'algorithme de décision. Une economie d'energie de l'ordre de 10% est réalisée. L'économie d'énergie peut sembler faible mais l'algorithme de décision permet aussi de limiter l'utilisation de la bande passante. En effet, le tableau 3.2 présente la quantité de données envoyée par le capteur pour chaque scénario. L'utilisation de la compression a permis de réduire le trafic généré par le capteur à 10% du trafic généré par le capteur en utilisation classique. Cette réduction est significative et permet en particuliers d'améliorer les performances du réseau. En effet, les capteurs Micaz et leur chipset utilisant le CSMA/CA comme méthode d'accès, le volume de données émis à donc un impact direct sur les performances du réseau. Cela permet alors d'améliorer la fiabilité du système.

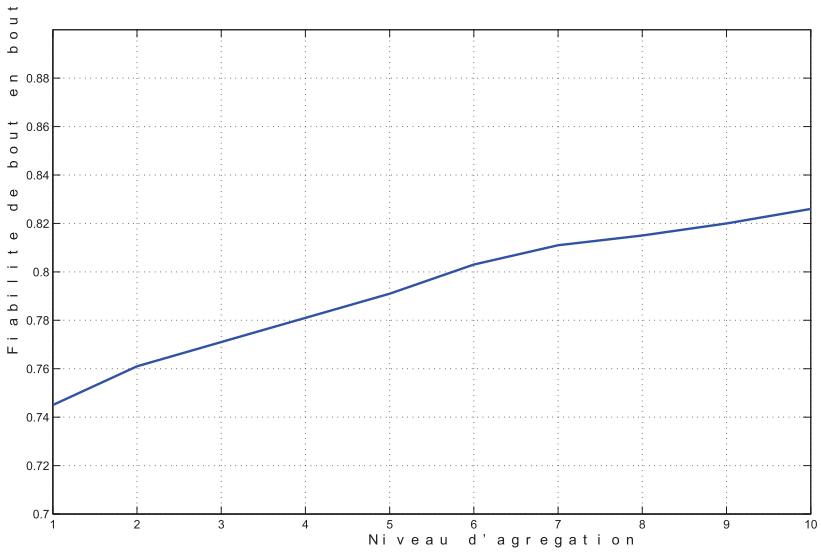

FIG. 3.8 – Fiabilité de bout-en-bout en fonction du degré d'agrégation des données [13]

La fiabilité des réseaux de capteurs est très importante et l'agrégation des données est une solution qui permet de l'améliorer [13, 17]. Dans [13], Benson *et al.* ont observé que la fiabilité de bout-en-bout est liée au degré d'agrégation des données.

Ils définissent la fiabilité de bout-en-bout comme étant le produit de la fiabilité de proche-en-proche de chaque liaison du chemin. Le degré d'agrégation est le nombre de messages agrégés dans un seul et même paquet. La fiabilité de proche-en-proche est défini comme étant la probabilité qu'un paquet soit reçu entre deux voisins. La figure 3.8 présente les résultats obtenus. On y voit clairement que la fiabilité est liée au degré d'agrégation et que, plus le nombre de messages agrégés est important, plus la fiabilité des communications augmente.

Les performances des réseaux de capteurs pour la médecine peuvent donc être améliorées par un algorithme de décision. L'utilisation de la compression permet d'économiser de l'énergie mais également de réduire l'utilisation de la bande passante et de fiabiliser les communications. Néammoins, la fiabilité passe également par un aspect sécurité. Les données médicales doivent être protégées. Il est donc nécessaire d'étudier le problème de la sécurité dans les réseaux de capteurs.

3.6 Conclusion

Dans ce premier développement de mécanisme permettant d'améliorer les performances de notre plate-forme de communications, nous avons développé et implémenté un algorithme d'ordonnancement des communications basé sur la sémantique des données à acheminer.

Cet algorithme est spécifiquement adapté au réseaux de capteurs pour la médecine. En effet dans notre solution, l'ordonnancement est réalisé en fonction des données générées par le capteur physique (*i.e.* l'équipement chargé de réaliser la mesure). Dans notre cas, nous avons considéré la constante médical ECG (ElectroCardioGramme). Cette constante représente le pouls du patient et necessite une surveillance particulière.

Considérant que dans la plupart des cas, les variations de cette données sont connu, il devient alors pertinent de développer une solution réduisant l'envoi des communications. Lorsque la donnée n'a pas évolué différement de la prévision, on peut choisir de ne pas l'envoyer au moment de la mesure mais attendre afin de pouvoir la compresser et ainsi réduire l'overhead induit par toutes communications réseaux. Dans le cas de réseaux sans fils dont les ressources sont limités, la réduction de cette overhead est essentielle. Cette réduction permet de réduire l'utilisation de la bande passante et de réduire la consommation d'enérgie pour l'envoi des données. Il faut néanmoins utiliser un surplus d'énergie pour compresser les données. L'énergie utilisé dans ce cas doit donc être inférieur à l'énergie qui aurait été utilisé pour envoyer l'ensemble des données sans compression.

Cette solution a était implémenté et testé sur des capteurs CrossBow utilisant le système d'exploitation TinyOS. La mesure d'ECG a été réalisé par l'usage d'un emulateur de prototype fourni par le laboratoire EPH de l'Institut Telecom SudParis.

L'ensemble des résultats permettent d'observer une nette réduction du nombre de paquets envoyés ainsi qu'une réduction de la consommation d'énergie au sein du capteur.

La réduction du nombre de paquets envoyés implique donc une meilleur utilisation de la bande passante disponible. La réduction du taux d'utilisation de bande passante pourrait permettre également d'améliorer la fiabilité du réseaux de capteur.

En effet, dans un réseau de capteurs utilisant une technologie ou la méthode d'accès au canal utilisée est concurentielle (*e.g.* CSMA), le taux d'utilisation de la bande passante est un paramètre essentiel. Une bande passante surchargée, ce sont des temps de latence plus importante et donc un retard dans les délais d'acheminement des données. Une bande passante libre réduit les temps de latence et assure donc des délais d'acheminement réduits et implicitement une meilleur réactivité du système et donc une meilleur fiabilité du réseaux.

La compression et l'envoi de ces données compressées est moins consommatrice d'énergie que les envois directs des données. Cette compression nécessite cependant de pouvoir bénéficier d'un espace de stockage sur le capteur. Le remplissage total de cette espage de stockage pouvant alors signifier la nécessité de l'envoi des données. Dans le cas d'une utilisation dans le domaine médical et donc dans notre étude, l'implémentation d'une *deadline* des données peut permettre de spécifier une date limite d'envoi des données. Et ceci assurant une périodicité controlée de la réception des données.

Chapitre 4

Implémentation de mécanismes de sécurité pour les réseaux de capteurs

La plate-forme que nous avons développé n'offre pas une sécurité suffisante. En effet, nous n'utilisons comme mécanismes de sécurité, uniquement une identification basé sur l'adresse MAC de l'équipement et sur la conaissance de mot de passe de 4 caractères. Il n'y a pas de mécanismes d'authentification des points d'accès et des utilisateurs et une attaque de type *Man-In-The-Middle* est alors aisément réalisable. Les données ne sont pas chiffrées et peuvent donc être lues et/ou altérées en clair par un tiers. Dans le domaine médical, cette sécurité des données est primordiale. Il est donc nécessaire de mettre en place une solution de sécurité plus élaborée permettant d'authentifier et l'utilisateur et la passerelle, mais également de chiffrer les données circulant sur le réseau.

Le problème de la sécurité dans les réseaux sans fils est une thématique très forte et très documentée. Dans le cadre qui nous intéresse, la sécurité relève d'une dimension particulière car elle doit être intégrée dans un contexte capteurs et donc avec des équipements ayant des ressources limitées. La sécurité, et plus particulièrement, le chiffrement nécessite des ressources importantes, notamment en terme de puissance de calcul.

L'architecture et les protocoles de sécurité proposés doivent alors être adaptés aux contraintes matérielles existantes.

4.1 Architecture

L'architecture que nous utilisons pour un réseau de capteurs pour la télémédecine est proche d'une architecture classique de surveillance [90] et proche de l'architecture

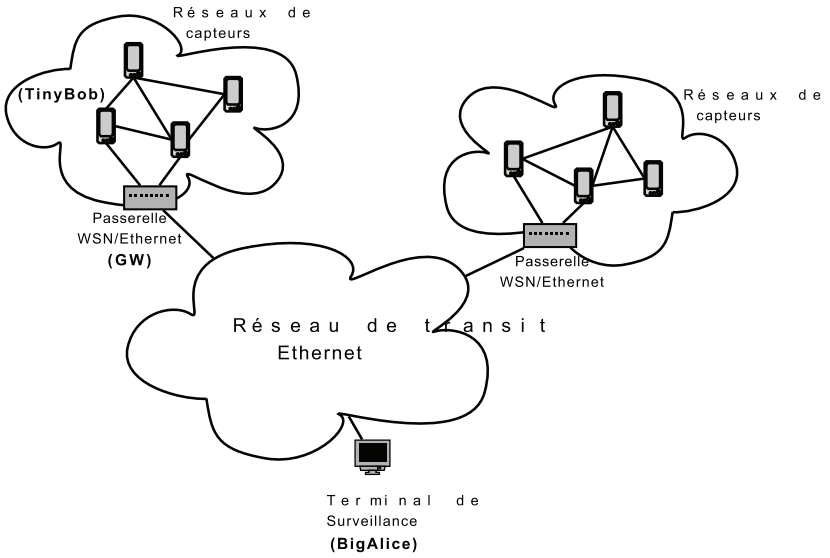

FIG. 4.1 – architecture de surveillance d'un réseaux de capteur médicaux

que nous avons présentée dans les sections précédentes. Un ensemble de capteurs communique entre eux et envoie des informations vers une passerelle. Cette dernière est reliée à un réseau de transit *classique* : filaire ou sans-fil. L'entité analysant les données est située sur ce réseau et les reçoit pour traitement (*cf.* Fig.4.1). La passerelle a pour rôle ici d'interfacer deux technologies différentes entre elles.

On suppose que le réseau de capteurs est normalisé IEEE 802.15.4 et que le réseau de transit est un réseau Ethernet. La passerelle doit donc modifier les messages depuis le réseau de capteurs et les convertir en trames Ethernet et paquets IP. Elle est également impliquée dans la politique de sécurité. Cette politique est basée sur le protocole TLS (Transport Layer Security protocol)[27].

TLS est un protocole qui permet de réaliser le chiffrement, l'authentification et le contrôle d'intégrité des données échangées entre un client et un serveur. TLS comporte plusieurs sous-protocoles, notamment un protocole de *Handshake*. Celui-ci permet à un client et à un serveur de négocier un cryptogramme, de l'authentifier et d'obtenir une clef principale partagée entre eux. Cela est réalisé en utilisant des algorithmes de clefs publiques. Une fois cette clef principale définie, les deux parties peuvent en dériver des clefs symétriques et ainsi chiffrer et authentifier des données.

Dans l'architecture proposée, les deux entités utilisent des technologies réseaux différentes. Il est donc nécessaire d'adapter le protocole TLS à cette architecture et aux contraintes capteurs. C'est dans ce contexte qu'a été développée la solution Tiny 3-TLS [33].

Algorithme	Temps(s)	taille des données (Octets)	taille du code (Octets)
ECC 160	0.81	282	3682
ECC 224	2.19	422	4812
RSA 1024 (pub)	0.43	542	1073
RSA 1024 (priv)	10.99	930	6292
RSA-2048 (pub)	1.94	1332	2854
RSA-2048 (priv)	83.26	1853	7736

TAB. 4.1 – Temps moyen d'exécutions des algorithmes ECC et RSA sur un Atmel ATmega128 [38]

4.2 Tiny 3-TLS

4.2.1 problématique

Tiny 3-TLS adapte le *handshake* de TLS dans le but d'établir un tunnel sécurisé de bout-en-bout entre un capteur et un terminal de surveillance. On distingue deux types d'architectures possibles. Dans le premier cas, la confiance envers la passerelle est partielle, c'est à dire que le message envoyé par le capteur vers la destination ne sera pas décodé par la passerelle. Cette dernière est uniquement utilisée pour le transport et la sécurisation des communications sur le réseau IP. Dans la deuxième architecture, la confiance dans la passerelle est totale. Elle pourra donc décoder le message venant du réseau de capteurs pour l'encoder à son tour set le transmettre au destinataire.

Ces deux architectures permettent de répondre à une application de surveillance médicale. Un médecin veut surveiller les constantes médicales d'un patient alors qu'il n'est pas à son bureau. Il lui est possible de se connecter sur la passerelle de l'hôpital et de récupérer les informations provenant des capteurs reliés au patient. La passerelle permet l'authentification des communications entre le terminal du médecin et les capteurs du patient, ainsi que leur chiffrement des données transitant sur chacun des réseaux.

Une clef secrète K est partagée entre la passerelle et le réseau de capteurs. Cette clef est utilisée afin de chiffrer les messages entre ces deux entités. Cette clef secrète est définie en utilisant la cryptographe sur les courbes elliptiques [65]. Cette méthode de chiffrement répond mieux aux contraintes capteurs car elle nécessite des clefs plus petites que pour les méthodes classiques de chiffrement de type RSA (*cf.* Tableau 4.1).

La syntaxe utilisée pour décrire Tiny 3-TLS est donnée dans le tableau 4.2.

$BigAlice$, $Passerelle\ GW$, $TinyBob$	Entités
K	Clef symétrique partagée entre la passerelle GW et le capteur TinyBob
PK_x	Clef publique de l'entité x
PK_x^{-1}	Clef privée de x
ID_x	Identifiant de l'entité x
$Cert_x$	Certificat de l'entité x
N_x	Nonce généré par x
P_x	Ciphersuite offert par x
$ECDH_x$	Valeurs publiques du Diffie-Hellman à courbe elliptique de x
$H(.)$	Fonction de hachage
$\{M\}\_K$	M chiffré avec la clef K
PMS	Pré-clef secrète
$A\|B$	Concaténation de A et B
M	Concaténation des messages envoyés entre BigAlice et la passerelle GW

TAB. 4.2 – Syntaxe de description de Tiny 3-TLS

4.2.2 Confiance partielle dans la passerelle

Avec cette architecture, la passerelle *GW* accepte que le capteur *TinyBob* et le terminal de surveillance *BigAlice* ait un secret partagé, sans pour autant le connaître. Cette architecture nécessite un protocole de communication plus complexe que dans le cas d'une architecture dans laquelle la confiance dans la passerelle est totale. Néammoins, elle permet aux informations d'être disponible en clair sur le terminal du médecin et les capteurs du patient. Le protocole de communication, basé sur un échange de clefs Diffie-Hellman, est décomposé en diverses étapes (cf Fig.4.2) .

1. BigAlice envoie un message *Client Hello* contenant son identifiant, l'identifiant de session, une offre de ciphersuite et un nonce.
2. Le message est chiffré avec la clef symétrique K et transféré par la passerelle GW vers TinyBob.
3. TinyBob renvoie un message *Server Hello* contenant son identifiant, l'identifiant de session, un nonce, une contre-offre de ciphersuite et ses valeurs publiques d'ECDH.
4. La passerelle conserve les valeurs d'ECDH pour elle-même et transmet à BigAlice un message *Server Hello* contenant l'identifiant de session, l'identifiant de TinyBob, un nonce et une contre-offre de ciphersuite. Elle transmet également son propre certificat et une requête de certificat.
5. BigAlice répond alors avec son certificat et ses valeurs d'ECDH publiques chiffrées avec la clef publique de la passerelle et ses valeurs d'ECDH publiques, le nonce de TinyBob et l'identifiant de session. Le tout est chiffré avec la clef privée de BigAlice.
6. La passerelle authentifie BigAlice et récupère ses valeurs d'ECDH publiques. Elle encode les valeurs d'ECDH publiques avec la clef publique de BigAlice et transmet le texte ainsi chiffré à BigAlice.
7. De la même façon, la passerelle envoie à TinyBob les valeurs d'ECDH publiques et un haché des messages reçus précédemment entre la passerelle et BigAlice. Le tout est chiffré avec K, la clef symétrique partagée entre la passerelle et TinyBob.
8. & 9. TinyBob et BigAlice échangent alors un message *Finished*. Le tunnel de chiffrement est mis en place et les deux entités peuvent maintenant communiquer ensemble.

Le message *Finished* est le suivant

$$Finished = H(R, M) \; ou \; R = PRF(DHK, N_{BigAlice}, N_{TinyBob}),$$

DHK est la clef de négociation $ECDH$.
$PRF \; Pseudo-Random \; Function$ est une fonction qui crée à partir d'une clef et d'une graine une chaine pseudo aléatoire.

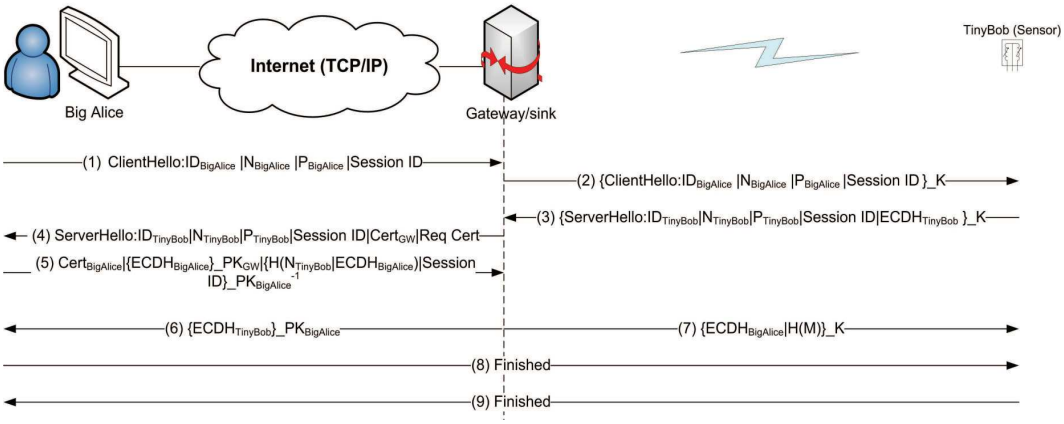

FIG. 4.2 – Etablissement de la politique de sécurité lorsque la confiance dans la passerelle est partielle

BigAlice envoie son message *Finished* chiffré avec la clef principale de BigAlice : *BigAliceMasterKey*. De la même façon, TinyBob envoie son message *Finished* chiffré avec la clef principale de TinyBob : *TinyBobMasterKey*.

$$BigAliceMasterKey = KeyGen(ID_{BigAlice}, N_{BigAlice}, N_{TinyBob}, R)$$
$$TinyBobMasterKey = KeyGen(ID_{TinyBob}, N_{BigAlice}, N_{TinyBob}, R)$$

4.2.3 Confiance totale dans la passerelle

Dans cette architecture, la confiance en la passerelle est totale. Cela signifie qu'elle est capable de décoder les messages venant de TinyBob et de BigAlice. Il existe donc un secret partagé entre ces 3 entités.

Le protocole de communication est, à l'instar de la solution précédente, basé sur un échange de clefs Diffie-Hellman, décomposé en diverses étapes (cf Fig.4.3). Les 4 premières étapes sont très proches de celles du scénario précédent.

1. BigAlice envoie un message *Client Hello* contenant son identifiant, l'identifiant de session, une offre de ciphersuite et un nonce.
2. Le message est chiffré avec la clef symétrique K et transféré par la passerelle GW vers TinyBob.
3. TinyBob renvoi un message *Server Hello* contenant son identifiant, l'identifiant de session, un nonce, une contre-offre de ciphersuite.
4. La passerelle conserve les valeurs d'ECDH pour elle-même et transmet à BigAlice un message *Server Hello* contenant l'identifiant de session, l'identifiant de TinyBob, un nonce et une contre-offre de ciphersuite. Elle transmet également son propre certificat et une requête de certificat.

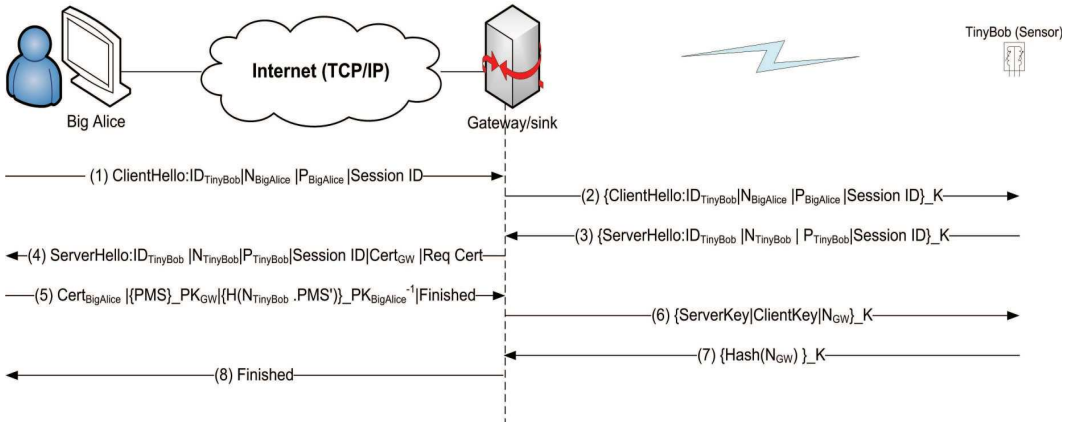

FIG. 4.3 – Etablissement de la politique de sécurité lorsque la confiance dans la passerelle est totale

5. BigAlice génère une pré-clef secrète symétrique(PMS). Elle envoie alors à la passerelle, son certificat, le PMS et le nonce de TinyBob, le tout étant chiffré avec la clef publique de la passerelle. BigAlice envoie également un message *Finished*.

$$Finished = H(R, H(M)) \; ou \; R = PRF(PMS, N_{BigAlice}, N_{TinyBob})$$

Une fois le message *Finished* de BigAlice reçu, la passerelle génère un nonce, une clef de lecture : *Client-read-key* et une clef d'écriture : *Client-write-key*.

$$Client - write - key = KeyGen(ID_{BigAlice}, N_{BigAlice}, N_{TinyBob}, R)$$
$$Client - read - key = KeyGen(ID_{TinyBob}, N_{BigAlice}, N_{TinyBob}, R)$$

6. La passerelle va alors chiffrer les 3 éléments générés avec la clef K et transmettre le résultat à TinyBob.
7. TinyBob va décoder le message reçu et renverra un haché de son identifiant et du nonce généré par la passerelle.
8. Enfin, la passerelle enverra un message *Finished* à BigAlice.

Ces deux architectures permettent à des noeuds situés dans le réseau de capteurs de communiquer avec un terminal de surveillance se trouvant sur un réseau technologiquement différent. Il est maintenant nécessaire d'évaluer leurs performances.

4.3 Analyse et Résultats

L'étude de ces architectures est réalisée en utilisant Avrora [11] et Avispa [10, 100]. Le premier nous permet de réaliser une étude quantitative en calculant avec précision les délais d'exécution des solutions proposées. Avispa est un outil permettant de valider

	Confiance partielle	Confiance Totale
Opérations supplémentaires	-1 chiffrement symétrique -1 déchiffrement symétrique	-1 chiffrement symétrique -1 déchiffrement symétrique
Opérations en moins	-1 vérification de signature	-1 génération de clefs partagées Diffie-Hellman -1 vérification de signature -Calcul des messages *Finished*

TAB. 4.3 – Comparaison des architectures de confiance partielle et totale avec la solution Sizzle

les protocoles de sécurité en se basant sur une description du protocole réalisé à l'aide d'un language de spécification de haut niveau (HLSPL). Cela nous permet alors de réaliser une étude qualitative des deux architectures proposées.

4.3.1 Performances

Afin de pouvoir quantifier les délais de chacune des architectures, Tiny 3-TLS a été développé en Nesc pour TinyOS 1.x. Cette implémentation utilise les librairies de courbes elliptiques disponibles. L'émulation via Avrora permet d'obtenir avec une très grande précision, les délais pour chacune des opérations nécessaires à l'établissement de la politique de sécurité. L'émulation a été réalisée en utilisant comme référence, un capteur Micaz [107].On rappelle que ce capteur utilise un microprocesseur ATmega128L de 8 bits cadencé à 7,37 Mhz. Le but de cette solution est de limiter le nombre d'opérations réalisé par le capteur, plus particulièrement les opérations cryptographiques sur les clefs publiques. Pour cela, certaines opérations sont déportées sur la passerelle, cette dernière n'étant pas, dans la plupart des scénarios, un équipement ayant des ressources limitées. Nous comparons ici les architectures proposées avec Sizzle [37]. Le tableau Tab.4.3 présente les différences, en termes d'opérations réalisées, avec Sizzle.

Dans le cas d'une confiance partielle dans la passerelle, on s'intéresse au déchiffrement symétrique du message *Hello* envoyé par la passerelle vers le capteur et au chiffrement de la réponse par ce dernier.

Le déchiffrement s'effectue en 44,8 ms et le chiffrement en 156,8 ms. Une vérification de signature par un capteur, même dans le cas d'une implémentation optimisée, sera plus longue d'au moins deux ordres de grandeur. Cela confirme que l'architecture pour laquelle la passerelle est partielle, est plus rapide et donc plus économique que la solution Sizzle.

De la même façon, lorsque la confiance dans la passerelle est totale, l'avantage de Tiny 3-TLS par rapport à Sizzle est indéniable. En effet, le chiffrement et le

déchiffrement s'effectuent alors en 44,8 ms dans chaque cas, le message encodé par le capteur ne contenant plus les élements publiques Diffie-Hellman à courbe elliptique (ECDH). Egalement, l'opération la plus *coûteuse*, la négociation de clef, est déléguée à la passerelle.

4.3.2 Validation

L'étude de performance faite, il est nécessaire maintenant d'analyser et de valider Tiny 3-TLS d'un point de vue sécurité. AVISPA (Automatic Validation of Internet Protocols and Application) permet, à l'aide du langage HLPSL (High Level Protocol Specification Language), de décrire un protocole et de connaître les objectifs de sécurité atteints. Le modèle d'intrusion utilisé est celui de Dolev-Yao [29] dans lequel toutes les communications avec l'intrus sont synchrones. En d'autres termes, l'attaquant a connaissance de tous les messages échangés entre les divers protagonistes. Il a également la capacité de réémettre des anciens messages (rejeu de paquets).

Dans le cas d'une confiance partielle, les objectifs de sécurité atteints sont :

- Authentification mutuelle forte entre la passerelle et BigAlice.
- Secret des clefs *TinyBobMasterKey* et *BigAliceMasterKey* entre BigAlice et TinyBob.
- Résistance au rejeu. Si un attaquant réémet des messages précédemment envoyés ou une session complète, il ne sera pas capable d'accéder au réseau.

Dans le cas d'une confiance totale, les objectifs de sécurité atteints sont :

- Authentification mutuelle forte entre la passerelle et BigAlice.
- Secret des clefs partagées *Client-write-key* et *Client-read-key* entre la passerelle, BigAlice et TinyBob.
- Résistance au rejeu de paquets.

4.4 Conclusion

La solution Tiny 3-TLS permet d'assurer une authentification entre la passerelle et BigAlice et également le secret des divers clef générées. La création d'un tunnel de bout-en-bout sécurisé entre un noeud et un terminal de surveillance est donc possible. Dans cette solution, la passerelle est pleinement utilisée pour permettre la création de ce tunnel.

L'introduction de la notion de confiance partielle ou totale permet de mettre en place cette solution dans différents types d'applications et d'environnement. On peut en effet, utiliser cette solution dans un milieu ou toutes les passerelles sont de confiances et ou elle possèdent toutes une clefs de chiffrement. Mais, Tiny 3-TLS permet également d'être utilisé dans le cas ou la confiance en la passerelle l'autorise à transmettre des données mais pas à en connaitre le contenu. Notre solution nous a également permis

de mettre en avant de meilleurs performances que la solution Sizzle[37] que nous avons présenté précédemment.

Dans notre plate-forme, cette solution se met facilement en place, les différents types d'entités se retrouvant dans les deux architectures. Il est alors possibles aux utilisateurs de s'authentifier et d'être sur que les passerelles sont de confiance. Les utilisateurs d'équipements Bluetooth peuvent alors échanger des messages et des données sans craindre qu'ils soient interceptés.

Conclusion

Nous avons présenté dans ce chapitre, une plate-forme de communication utilisant la technologie Bluetooth. Cette plate-forme permet à deux personnes étant physiquement éloignée l'une de l'autre de communiquer ensemble via un équipement Bluetooth et une infrastructure. Cette plate-forme s'inspire des réseaux cellulaires classiques (*e.g.* GSM). Sa particularité est l'utilisation de la technologie Bluetooth pour la communication au sein de chaque cellule. Cette technologie était initialement conçu pour remplacer les connectiques informatiques entre les différents équipements : ordinateur, souris, téléphone portable. Dans le développement de notre plate-forme, nous avons utilisé cette technologie afin de transmettre des données et des messages entre les utilisateurs.

Bien que l'échange de données et de messages puissent semblé proche des informations que les équipements Bluetooth sont amenés à échanger, la complexité de notre plate-forme a nécessité le developpement de mécanismes spécifiques. Ceci afin de répondre aux exigences du projet : gestion de la mobilité des équipements, du roaming entre les différentes cellules, de l'authentifications des usagers.

L'analyse et la modélisation de la plate-forme nous ont permis de spécifier ses limites. Ceci nous permet alors de définir certaines régles d'implantations qui permettent d'aider à l'installation de la plate-forme dans différents lieux. Il faut cependant préciser que certaines caractéristiques du modèle tels que la loi de probabilité suivi pour le processus d'arrivée des personnes (ici loi exponentielle), peut ne pas être adapté à tous les scénarios.

Les experimentations que nous avons effectué nous on cependant permis de mettre en évidence les faiblesses de notre plate-forme et de la technologie Bluetooth. La principale faiblesse est tout d'abord l'aspect sécurité de notre plate-forme et des plate-formes de réseaux personnel : *WPAN* de façon générale. Il n'existe pas de mécanisme fiable d'authentification dans la technologie Bluetooth et intrinséquement dans notre plate-forme. Il en va de même pour les plate-formes "capteurs". C'est pourquoi nous avons développé un protocole adapté aux réseaux de capteurs : Tiny 3-TLS [33] permettant de chiffrer les messages et d'authentifier une plate-forme de communication. Cette solution est basée sur le protocole TLS (Transport Layer Security protocol)[27]. Elle permet à des équipements ayant des ressources limitées (*e.g.* capteur) de communiquer de façon sécurisé. Dans notre cas, elle pourrait être utilisé entre les équipements Bluetooth, les

points d'accès Bluetooth et le serveur central afin d'authentifier ce dernier et de chiffrer les communications Bluetooth de manière efficace.

Cette expérimentation n'a put être réalisé pour ce projet car aurait nécessité le développement complexe de mécanismes sur plate-forme téléphonique mobile équipée Bluetooth ce qui est en dehors de notre domaine de compétence.

Nous avons également pu mettre en évidence des insuffisances en termes de consommation d'énergie et de bande passante. L'utilisation d'un algorithme de décision [35] basé sur la compression et la qualification des données permettrait de réduire la consommation d'énergie et de limiter l'utilisation de la bande passante. Cette solution utilise un algorithme de compression non destructif permettant de réduire le volume de données à transmettre. Cette compression utilisant moins d'énergie qu'une transmission, elle permet de réduire l'énergie nécessaire à l'envoi d'une donnée. Cette réduction permet alors d'économiser de la bande passante.

L'aspect fiabilité des communications est également un aspect de notre plate-forme qui pourrait être amélioré. Dans le cas d'une utilisation de cette plate-forme dans un milieu médical, cette fiabilité est essentielle. Dans la suite de ce mémoire, nous présenterons des solutions permettant d'apporter la fiabilité nécessaire dans un réseaux de capteurs.

Deuxième partie

Fiabilité des réseaux de capteurs

La fiabilité est une problématique très importante dans les réseaux filaire et sans-fils. Elle l'est encore plus dans le domaine des réseaux de capteurs. Dans ce type de réseaux, les équipements ont des ressources limités, et peuvent évoluer dans des environnement perturbés. Ces particularités imposent donc que les mécanismes classiques permettant de fiabiliser un réseau ne doivent être utilisé pour les réseaux de capteurs. Le traitement de la fiabilité, à l'instar du routage, des protocoles de couches MAC ou des applications doit bénéficier d'un traitement particulier propre aux contraintes des réseaux de capteurs.
L'utilisation de métrique plus adaptée aux réseaux sans fils est une possible solution quand à l'amélioration de la fiabilité. L'utilisation d'information qualifiant le lien reliant deux noeuds entre eux pourrait permettre de qualifier la fiabilité de ce lien. Les nouvelles normes définisse ce genre de métrique et les équipements fabriquées par les différentes sont maintenant capables de fournir ce genre d'information. Dans ce chapitre, nous commencerons par expliciter ce qu'est la fiabilité et ce qu'elle a de particulier dans les réseaux de capteurs. Nous présenterons ensuite les solutions que nous avons développé permettant d'améliorer la fiabilité des réseaux de capteurs. Nous présenterons tout d'abord une solution (WSC-MAC) basée au niveau de la couche liaison du modèle OSI des réseaux de capteurs. Cette solution utilise la coopération pour permettre un taux de réception des paquets plus important. Nous présenterons ensuite une solution (MAODV-SIM) utilisant le principe de route de secours au niveaux réseaux du modèle OSI. Cette solution, à l'instar de WSC-MAC permet d'améliorer la réception de paquets.

Qu'est ce que la fiabilité

Si on se réfère à la définition IEEE [43], la fiabilité est *la capacité d'un système ou d'un composant à réaliser ses fonctions définis par des conditions initial pendant un laps de temps.* Concrètement, un système est fiable aussi longtemps qu'il réalise la tâche qu'il lui incombe.

Les réseaux de capteurs sont composés de capteurs que l'on peut considérer comme étant des équipements "fragiles", ce qui rend implicitement ces réseaux fragile.

La notion de défaillance de terminaux est un paramètre peu courant dans les réseaux sans fils et encore moins dans les réseaux filaires classiques. En effet, dans un réseaux de capteurs, chaque terminal est lui même un routeur et responsable de données qui ne lui sont pas destiné. Sa défection entraine donc une possible diminution de la connectivité du réseaux et des pertes de paquets. Dans un réseau classique, tous les terminaux sont en général considérés comme fiable et toujours disponibles. Dans un réseau de capteurs ceci n'est pas vrai. De par leur conception et l'environnement dans lesquels ils peuvent évoluer, les capteurs peuvent être amenés à ne plus fonctionner. En effet, de nombreux projets sont utilisés dans des environnements très inhospitaliés : volcans [7], glaciers [91], forêt [84], surveillance urbaine [69].

Ces différents déploiements montrent aussi des rendements pouvant être éloignés des résultats initialement prévu (cf. Fig.4.4)

C'est à dire que certains noeuds se trouvent inaccessible et donc la surveillance de l'environnement initialement planifié n'est pas obtenu. Dans certains cas, les analyses ultérieurs ont permis de définir les causes de ces défaillances : réseaux, logiciels, conflits radio etc... Mais il est aussi des défaillances qu'on ne peut expliquer ou qui resulte d'incidents physiques (*e.g.* destruction).

Barrenetxea *et al.* [12] ont analysé différents déploiements de réseaux de capteurs dans des environnements très divers. Ils ont identifié trois principales taches qui permettent d'aboutir à un déploiement performant :

- Développement. Dans cette première partie, la plus importante, l'environnement final dans lequel la plate-forme va être déployé doit déjà être pris en compte. Les équipements doivent notamment être résistant au conditions climatiques qu'ils peuvent rencontrer. Le développement logiciel de l'application doit également être exempt de "bugs" [19].

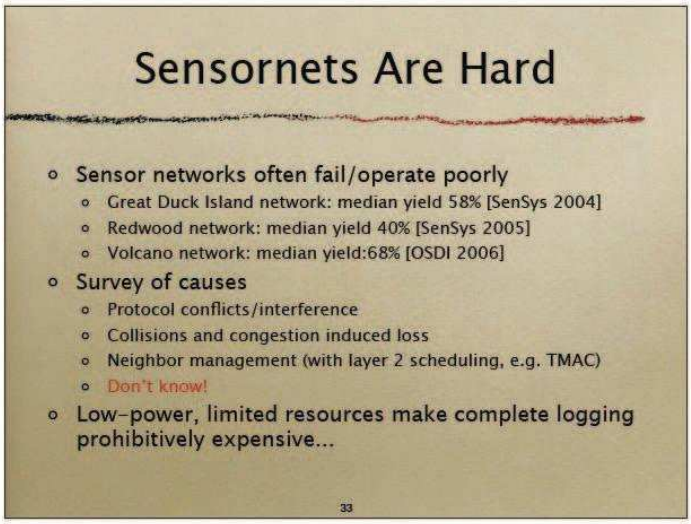

FIG. 4.4 – Rendement de différents déploiements de réseaux de capteur [54]

- Tests. Les tests réels en environnement contrôlé ne doivent pas être négligé. Ils peuvent notamment permettre de déceler des défauts de conceptions logiciels. Ils permettent également de valider dans une certaine limite, les choix matériels effectués dans la phase de développement.
- Deploiement. Le déploiement en lui même est souvent l'étape qui confirme ou infirme les différents choix effectués dans les précédentes étapes. Le déploiement physique en lui-même doit être donc considéré comme une part importante du projet global et être considéré tout au long du projet.

Ces différents travaux montrent que contrairement à un réseau classique, la possible défection des capteurs doit être pris en compte dans le développement des protocoles et des solutions mises en place.

Les mécanismes développés pour améliorer la fiabilité doivent donc prendre en compte la notion de défaillance possible de noeuds et de la possible fragilité du réseau.

Les métriques de performances classique des réseaux : débit, délai de bout-en-bout, temps de latence bien que toujours importantes, sont dans ce domaine moins contraignantes. La métrique délai de bout-en-bout peut être considéré dans certains scénarios, comme nous le verrons dans la suite de ce document, comme une métrique de fiabilité. Dans notre étude, la fiabilité est correlée au taux de réception de paquets. C'est une *métrique* qui ne peut être négligée et qui doit bénéficier d'optimisations et de mécanismes appropriés.

Chapitre 5

Communications coopératives pour les réseaux de capteurs

Afin d'améliorer la fiabilité des réseaux de capteurs, nous avons donc basé notre étude sur l'augmentation du nombre de paquets reçus. Pour cela, nous nous sommes notamment concentrés sur des techniques de communications coopératives qui assurent une meilleur réception des paquets.

En nous basant sur les spécifités des protocoles MAC présentées précédemment, nous avons développé une solution de communications cooperatives adaptée aux réseaux de capteurs [63].

La coopération est une technique prometteuse qui permet d'améliorées les performances des réseaux sans fils. Dans le cadre d'un réseau de capteurs, cette technique peut également être utilisée et les performances s'en trouvent alors améliorés. Néammoins, les solutions existantes sont basées sur des technologies sans fils ayant des ressources plus importantes (*e.g.* 802.11abg). Ces solutions ne sont pas adaptées aux contraintes des réseaux de capteurs. La définition du noeud relayant l'information est un des premiers problème à résoudre. Dans un réseau sans fils ayant des ressources "illimitées", il est possible d'établir un mécanisme d'échange de messages permettant de définir qui sera ce noeud relais. Dans un réseau de capteur, l'energie est limitée et on ne peut donc pas l'utiliser "inutilement". La solution WSC-MAC permet de définir un noeud relais en limitant très fortement les échanges de messages.

5.1 WSC-MAC : un protocole MAC coopératif

Cette solution est basée sur un algorithme utilisant le niveau d'énergie du signal reçu et un identifiant de groupe défini préalablement. Ceci permet à un noeud de sélectionner un noeud parmi ses voisins afin que celui-ci participe à la transmission des données. Cette solution permet de limiter l'utilisation de messages permettant de déterminer le noeud relais. En effet, la problématique de la coopération dans les réseaux sans fils

est de définir le noeud qui sera chargé de relayer l'information vers le noeud final. WSC-MAC propose un mécanisme réduisant significativement les messages nécessaires à cette détermination.

5.2 Sélection automatique du relais

Dans un réseau coopératif, un même message est transmis par plusieurs noeuds en même temps, sur le même canal, vers le destinataire. Afin d'atteindre le compromis entre le gain apporté par la coopération et la surcharge de trafic induite, Fan *et al.* [32] et Moh *et al.*[66] ont démontré que la "meilleure" coopération se produit lorsque peu de noeuds sont mis en oeuvre. En d'autres termes, plus le nombre de noeuds relais entre un expéditeur et un destinataire est faible, meilleure est la coopération.

La définition d'un algorithme permettant de limiter le nombre de noeuds est donc nécessaire. Cette algorithme doit également respecter les contraintes des réseaux de capteurs. En se basant sur ces observations, nous avons développé un algorithme qui permet à un noeud envoyant un paquet de "sélectionner" un petit nombre de noeuds qui devra alors envoyer ce paquet vers le destinataire.

Cette selection est réalisée à l'aide d'un nouvel identifiant (un identifiant de groupe) : $Group\_Id$. Celui-ci est défini lors du processus d'autoconfiguration de chaque noeud. Il diffère de l'adresse réseau du noeud mais il ne doit pas être unique dans le réseau.

L'algorithme 1 présente le mécanisme permettant à chaque noeud de définir son propre identifiant de groupe. Lors du processus d'autoconfiguration, chaque noeud va choisir aléatoirement un nombre entre 0 et A, le nombre moyen de voisins dans le réseau. Ce nombre deviendra alors son identifiant de groupe et ne changera plus pendant toute la durée de vie du réseau. La valeur A est calculée à partir de la taille du réseau, de la portée de chaque noeud et du nombre de noeuds composant le réseau. L'utilisation de cet identifiant permet à l'algorithme de coopération de réduire le nombre de relais lors d'une communication. Afin d'optimiser au maximum les performances de notre solution, deux noeuds voisins ne doivent pas avoir le même identifiant de groupe. Par contre, un noeud peut avoir deux voisins ayant le même identifiant de groupe. Le mécanisme d'autoconfiguration d'un noeud doit prendre en compte les identifiants de groupe de chaque voisin lors de la détermination de son propre identifiant. Dans notre algorithme, A_list est une liste de A valeurs disponibles.

Lors de l'envoi d'un paquet, le noeud source va choisir aléatoirement un identifiant de groupe et l'inclure dans l'en-tête du paquet. Il enverra ensuite le paquet vers son destinataire. Chaque noeud recevant ce paquet va comparer l'identifiant de groupe contenu dans le paquet avec son propre identifiant. Si les deux concordent, il doit participer à la communication ; dans le cas contraire, il ne participe pas à la communication et doit détruire le paquet. Dans la figure fig.5.1, le noeud S souhaite envoyer un paquet au

Input: A : nombre moyen de voisin
Output: my_group_ID

create *A_list* using *A*;
my_group_ID = 0;
while *(my_group_ID = 0)* **do**
  Ecoute du canal pendant un temps défini aléatoirement;
  if *(un voisin envoi son Group_ID)* **then**
    Cet identifiant doit être retiré de la liste A;
    $A\_list = A\_list - Group\_ID$;
  else
    Je me choisis un identifiant parmi la liste A;
    if $(A\_list = NULL)$ **then**
      my_group_ID = random(A_list);
    else
      creation d'une *A_list* à partir de *A*;
      my_group_ID = random(A_list);
    end
    Diffusion(my_group_ID);
  end
end

Algorithm 1: Algorithme permettant à chaque noeud de définir son identifiant de groupe

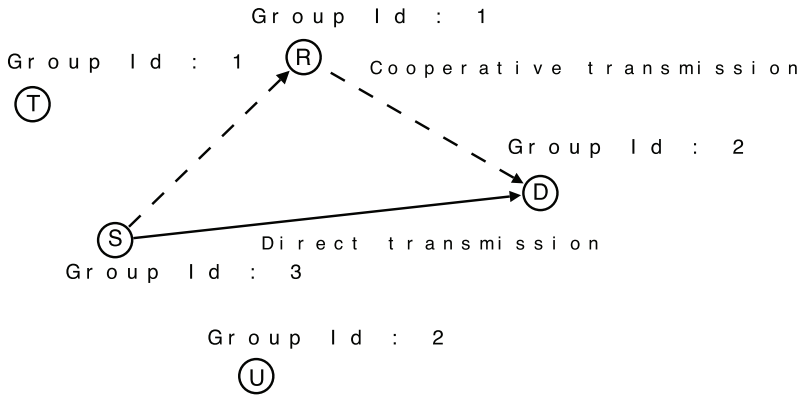

FIG. 5.1 – Scénario de communication coopérative

noeud D. Il décide pour cette transmission d'utiliser la coopération de ses voisins. Il choisit aléatoirement un identifiant de groupe pour ce paquet, ici 1. Les noeuds R et T ayant cette valeur comme identifiant de groupe, doivent participer à la communication en transmettant à leur tour le paquet vers la destination. Néammoins, afin de ne pas dépenser inutilement de l'énergie, chaque noeud appartenant au groupe désigné doit évaluer la qualité de son lien avec la destination et déterminer si sa propre transmission apporterait une amélioration à la transmission coopérative. C'est pourquoi nous avons optimisé notre solution en y ajoutant un deuxième algorithme.

5.2.1 Evaluation du lien

Dans la section précédente, nous avons présenté une solution distribuée permettant de définir un relais parmi les voisins d'un noeud désirant envoyer un paquet. Cette coopération doit intervenir si la transmission du noeud relais permet d'améliorer la communication. Il est donc nécessaire de développer un algorithme de décision qui permettant de définir ces noeuds relais. Les communications dans un réseau de capteurs étant grandes consommatrices d'énergie, le mécanisme de décision devra être local et exécuté par chaque noeud relais. Cet algorithme permettra au noeud, selectionné au préalable par l'intermédiaire de l'identifiant de groupe, de determiner s'il intervient dans la transmission. Pour cela, le noeud estime la qualité du lien qu'il a avec le destinataire du paquet et la compare avec la qualité du lien entre la source et la destination. Cela impose que chaque noeud est une table d'état de lien contenant la qualité du lien vers chacun de ses voisins. Le stockage disponible dans un noeud étant limité, cette table d'état de lien peut être partagé avec la table de routage de la couche réseau. Cette dernière comporte en général l'ensemble de ces voisins et permet de limiter l'utilisation des ressources du noeud. Si la qualité du lien du relais avec la destination est meilleure que la qualité du lien de la source avec la destination, il doit alors participer à la communication. L'estimation de la qualité du canal LQI(Link Quality Indication) est réalisée à partir de la valeur $RSSI$ (Receive Signal Strength Indicator) obtenue de la couche physique. Le $RSSI$ est mesuré par les équipements 802.15.4. Lorsqu'un noeud reçoit un paquet de données depuis un voisin. Le *chipset* radio fournit en plus des données, la qualité du signal reçu. Cette valeur nous permet alors de qualifiée la qualité du lien avec l'expéditeur. La mise à jour de la table d'état de lien de chaque noeud est réalisée à chaque communications reçues.

Reprenons l'exemple de la figure Fig.5.1, les noeuds R et T, après leur selection par S (en verifiant le $Group\_Id$ embarqué dans le paquet) doivent vérifier si leur transmission est bénéfique à la coopération. Le Noeud R vérifie la valeur LQI de son lien avec D dans sa table d'état de lien, calcule la capacité du canal à partir du rapport signal à bruit SNR et décide ou non de transmettre le paquet vers D.

Pour définir l'algorithme d'état de lien et le LQI, nous avons basé notre solution sur les caractéristiques de la couche physique des réseaux de capteurs utilisant le standard

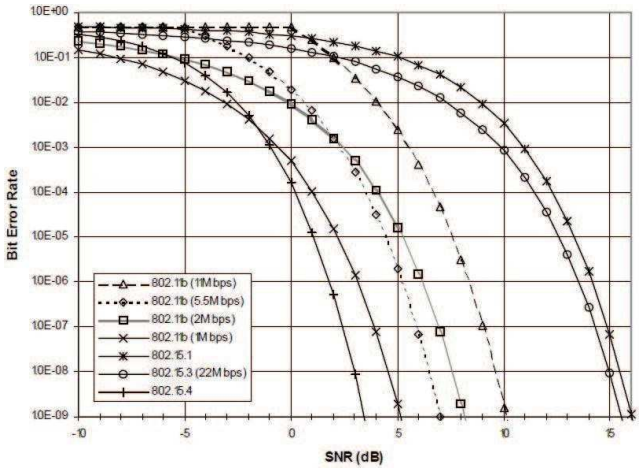

FIG. 5.2 – Bit Error Rate en fonction du SNR(dB) du signal reçu [44]

IEEE 802.15.4 [44] pour récupérer le "bit error rate"(*cf.* Fig.5.2). Nous avons également utilisé les travaux de Laneman *et al.* [51] pour le Space-Time-Coded Cooperative Diversity.

En se basant sur ces informations, un noeud relais est capable d'estimer la qualité d'une transmission entre deux voisins avec et sans coopération. Considérant, le Maximum Ratio Combining Diversity [16] (MRC), le rapport signal à bruit de la transmission coopérative est égale à la somme de chacune des transmissions qui la compose. On obtient la relation donnée par l'équation Eqn.5.1 où S_r est le signal total reçu à l'arrivée, N le nombre de relais et S_j le signal reçu depuis le noeud j.

$$S_r = \sum_{j=0}^{N} S_j \tag{5.1}$$

Le rapport signal à bruit ainsi obtenu permet de déterminer le *bit error rate* de la transmission coopérative. Pour le calcul de la capacité du canal en utilisant la transmission coopérative, on somme les informations mutuelles pour obtenir l'équation Eqn.5.2 où W est la bande passante comme définie dans la norme [44], SNR_rd est le SNR entre le relais et la destination et SNR_sd est le SNR entre la source et la destination.

$$\mathrm{Capa}_{\mathrm{coop}} = W \cdot \frac{1}{2}[\log(1 + \mathrm{SNR}_{\mathrm{rd}}) + \log(1 + \mathrm{SNR}_{\mathrm{sd}})] \tag{5.2}$$

A partir de la formule donnée dans l'équation Eqn.5.2 et des informations stockées par le noeud relais *LQI*, celui-ci est capable de prendre la décision de transférer le paquet vers le destinataire. L'algorithme 3, présenté dans la section suivante, présente le processus exécuté lors de la réception d'un paquet par un noeud du réseau.

5.2.2 Détails protocolaire de WSC-MAC

Nous avons basé notre solution sur une couche MAC de type CSMA adaptée au réseau de capteurs : BMAC[78]. Cette solution utilise des mécanismes permettant de réduire au maximum la période d'activité du noeud. Pour réduire la consommation d'énergie dûe à l'écoute oisive (*idle-listening*), BMAC implémente le principe de préambule pour informer qu'un paquet va être envoyé [78, 18]. Dans notre cas, ce mécanisme nous permet de synchroniser les noeuds et d'être sûr que les potentiels noeuds relais sont actifs lors de l'envoi du paquet.

La figure 5.3 présente la séquence d'envoi de trame de notre protocole.

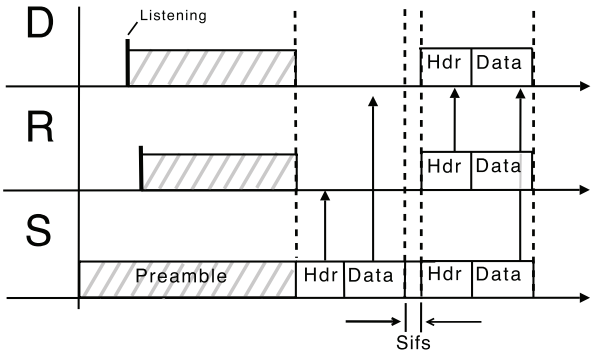

FIG. 5.3 – Séquence d'envoi d'une trame entre une source S, un relais R et une destination D

Node S utilise un préambule pour la synchronisation avec ses voisins. Puis le mécanisme de coopération se déroule en deux étapes. La première est la transmission du paquet de la source à la destination. Tous les noeuds entendant ce message doivent vérifier s'ils doivent ou non le transférer (en utilisant les algorithmes précédents). La seconde étape est l'envoi de ce même paquet en utilisant le Space-Time-Coded Cooperative Diversity. Celui-ci doit être envoyé par la source et le relais en même temps pour bénéficier de la coopération [16]. Pour synchroniser les envois de la source et du relais, on rajoute un délai SIFS entre la réception du paquet par le relais et son transfert. Ce délai permet au relais d'exécuter l'algorithme de décision. Le noeud source attend également un délai entre les deux envois successifs du paquet.

L'algorithme 3 présente le mécanisme exécuté par chaque noeud du réseau lors de la réception d'un paquet. L'algorithme 2 présente le mécanisme exécuté par le noeud source pour l'envoi d'un paquet. Bien que la coopération soit bénéfique dans de nombreux cas, elle n'est pas toujours nécessaire. En effet, si la source et la destination sont proches, le paquet sera parfaitement reçu sans coopération. Plus la distance entre les deux noeuds est grande, plus la coopération est intéressante. Le noeud source a la possibilité décider d'utiliser ou non de la coopération. Un seuil de décision est défini, il

permet de définir une qualité de lien en dessous de laquelle, le paquet a un pourcentage non négligeable de perte. Si le noeud source décide de ne pas utiliser la coopération, il positionne la variable $Group\_Id$ à 0. S'il utilise la coopération, la variable $Group\_Id$ est alors utilisée normalement et positionnée à une valeur définie aléatoirement (*cf.* Alg.2). Lors de la réception d'un paquet, chaque noeud, s'il n'est pas le destinataire, vérifie les champs Group_Id et Link_Quality du paquet pour vérifier leur implication ou non dans la communication (*cf.* Alg.2).

Input: Link-table
Output: Packet PktI

Prepare the packet PktI for sending to node x;

if *(Link-table.x.Link-State>seuil)* **then**
  Le paquet n'a pas besoin d'être transféré;
  PktI.Group_ID = 0;
  Send Packet PktI;
else
  Le paquet doit être transféré;
  PktI.Link_Quality= Link-table.x.Link-State;
  PktI.Group_Id = valeur aléatoire;
end
Send Packet PktI;
wait SIFS;
Send Packet PktI;

Algorithm 2: Algorithme effectué sur chaque noeud lors d'un envoi de paquet

Si un noeud n'est pas impliqué dans la communication, il détruit le paquet et retourne en phase de sommeil. S'il est impliqué et doit transférer le paquet, il attend un SIFS et envoie le paquet au destinataire. La source, si elle décide d'utiliser une transmission coopérative pour plus de fiabilité, envoie le paquet une première fois, attend un temps SIFS, et retransmet de nouveau le même paquet.

A destination, le noeud reçoit le premier paquet et détermine s'il doit ou non attendre la deuxième transmission. Si le paquet est corrompu et si le champ Group_Id est différent de 0, le destinataire sait qu'une nouvelle transmission doit intervenir. Si le paquet est corrompu et que le champ Group_Id est égal à 0, la transmission est incomplète et il n'y aura pas de nouvelle transmission.

Dans ce dernier cas, le comportement de la source et de la destination dépend du type d'acquittement mis en place. Dans notre cas, nous avons implémenté deux types d'acquittement : ACK et NACK. ACK correspond à un acquittement classique : chaque paquet reçu doit être acquitté. Si l'acquittement n'est pas reçu par la source à expiration du *timeout*, une nouvelle séquence de transmission est mise en oeuvre. NACK

Input: Packet PktI
Output: Packet PktI

reçoit un paquet;

```
if (PktI.MAC_Address = my address) then
    if (paquet est illisible) ET (PktI.Group_Id = 0) then
        attendre la transmission cooperative;
    else
        paquet est lisible;
        Passer le paquet à la couche supérieur;
        Retour en mode sommeil;
    end
else
    if (PktI.Group_Id ≠ 0) then
        if (PktI.Group_Id = my Group_Id) then
            if (PktI.Link_Quality>Link-table.(PktI.MAC_Address.Link_State)
            then
                La qualité de lien jusqu'à la destination est moins bonne que celle
                de la source;
                Jeter le paquet;
            else
                Attendre un temps SIFS;
                Envoyer PktI;
            end
        else
            Jeter PktI ;
        end
    else
        Jeter PktI ;
    end
    Retour en mode sommeil;
end
```

Algorithm 3: Algorithme effectué sur chaque noeud recevant un paquet

correspond à un acquittement sur non réception. Si le paquet est reçu mais corrompu et s'il n'y a pas de transmission coopérative ou si elle échoue également, le destinataire enverra alors un NACK à la source. A la réception du NACK, la source redémarre une séquence de transmission. On ne considère pas la perte totale d'un paquet (*i.e.* aucune réception à destination) dans notre étude. La perte totale d'un paquet est traitée par les couches supérieures qui peuvent solliciter l'expédition d'un nouveau paquet.

5.3 Analyse et performances de WSC-MAC

Nous présenterons dans cette section les résultats obtenus et montrerons que les communications coopératives et notre solution en particulier permettent d'améliorer les performances des réseaux de capteurs. Afin de quantifier ces performances, nous avons développé un simulateur réseau sous Matlab [79] utilisant notre solution.

Notre simulateur implémente le modèle de communications coopératives défini par Laneman *et al.* [50, 51] en plus de notre solution (*cf.* Alg.3). Chaque point d'une courbe représente la moyenne de 450 simulations (30 itérations pour 15 topologies différentes)d'un réseau contenant 100 noeuds. L'intervalle de confiance est ici de 95%. Pour valider nos résultats, nous les avons comparés à des transmissions directes n'utilisant pas de coopérations. Les paramètres de simulations restent identiques. Pour des raisons de clarté, nous avons résumé les paramètres de simulations dans la table 5.3.

Paramètres	Valeur
Paramètres de Simulations	
Nombre de noeuds	100
Densité de noeuds (noeuds/m^2)	0,01 - 0,25
Nombre de topologies	15
Nombre d'itérations	30
Paramètres MAC/PHY	
Portée de transmission (mètres)	35
Facteur d'atténuation	3
Modèle d'affaiblissement	Free Space
Modèle Physique	802.15.4
Sensibilité à la reception (dbm)	-90
taille du paquet de données (bits)	200
Taille des paquets d'acquittements (ACK/NACK) (bits)	40

TAB. 5.1 – Paramètres de Simulations

Dans cette section, nous mettrons en avant les différents aspects de notre solution en fonction de la densité du réseau ($noeuds/m^2$). En effet, les performances de notre

protocole sont liées à la probabilité qu'une communication coopérative se mette en place. Et la probabilité qu'un noeud puisse devenir relais d'une communication est fortement lié à la densité du réseau.

Taux de paquets reçus Afin de quantifier notre solution, le taux de paquets reçus ne prend pas en compte les paquets envoyés par les relais. Pour ce taux, nous avons calculer le rapport entre le nombre de paquets envoyés par la source et le nombre de paquets reçus par la destination. Une transmission coopérative est composée de deux envoies successifs, mais un seul envoi sera considéré, celui de la transmission coopérative. Pour la transmission directe, le paquet sera donc envoyé deux fois.

La figure Fig.5.4 montre le taux de paquets reçus (*PDR -Packet Delivery Ratio-*) en fonction de la densité du réseau. On peut observer une amélioration de l'ordre de 10% lorsque le réseau n'est pas dense ($< 0,1 noeud/m^2$). Pour une densité supérieure, les performances de notre solution sont proches d'une transmission directe sans coopération. En effet, dans ce cas, les distances entre les noeuds sont plus faibles. Le rapport signal à bruit augmente, améliorant le bit error rate et donc la transmission.

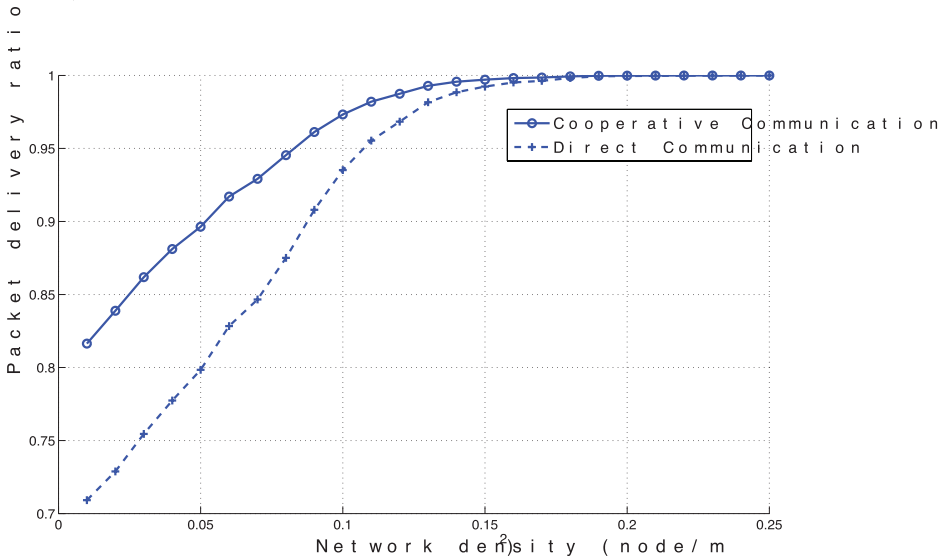

FIG. 5.4 – Taux de paquets reçus en fonction de la densité du réseau

Fiabilité Pour quantifier la fiabilité de notre solution, nous utilisons les deux types d'acquittement présentés précédemment : ACK et NACK. A l'instar du taux de paquets reçus, notre solution est plus performante lorsque la densité est faible (cf Fig.5.5). Dans un réseau dense, les liaisons sont de meilleurs qualité et les probabilité de pertes réduite. La probabilité de retransmission est donc également réduite. Dans le cas d'une

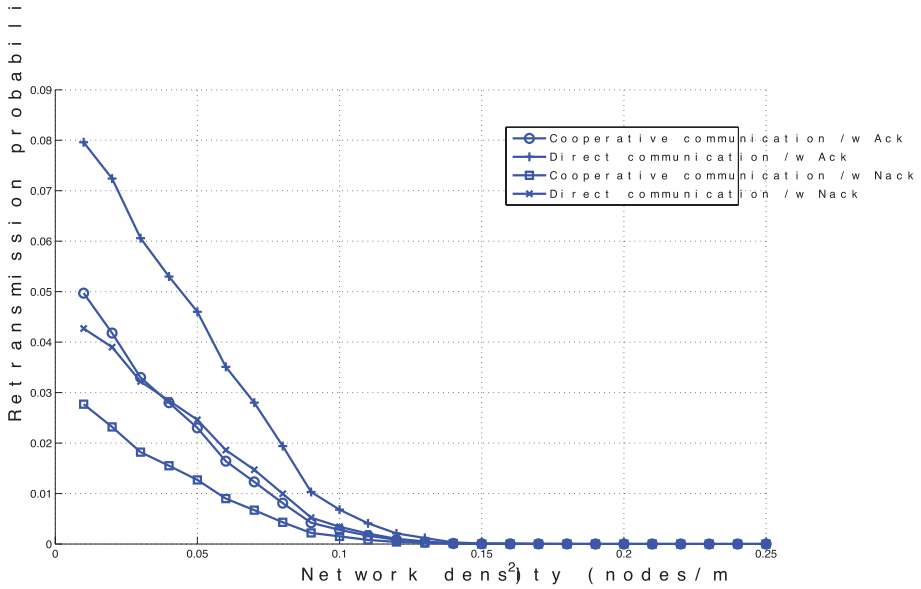

FIG. 5.5 – Probabilité de retransmissions (avec ACK et NACK) en fonction de la densité du réseau

utilisation des acquittements ACK, la probabilité de retransmission est réduite de 35% par l'utilisation des transmissions coopérative. Dans le cas d'une utilisation des acquittements NACK, la probabilité de retransmission est réduite de 40%. L'utilisation conjointe de la coopération et du NACK, en comparaison d'une communication directe utilisant les acquittements ACK, permet de réduire la probabilité de retransmission de 65%.

L'utilisation d'aquittement négatif permet de réduire la probabilité de retransmission. En effet, l'aquittement négatif n'est utilisé que si on détecte une perte de paquet. L'acquittement classique renverra le paquet tant que ce dernier ne sera pas acquitté. Si la liaison n'est pas fiable, un même paquet peut être envoyé plusieurs fois avant d'atteindre correctement sa destination et donc engendrer des retransmissions tant que l'acquittement n'est pas reçu. Néammoins, l'utilisation de l'aquittement négatif nécessite certains mécanismes. Il faut que le protocole soit capable de demander la réémission d'un paquet s'il est incorrectement reçu. Il faut également un mécanisme de détection de paquet perdu avertissant le protocole de couche MAC qu'un paquet est absent.

Capacité réseau La figure Fig.5.6 montre la capacité moyenne du canal en fonction de la densité. Elle est obtenue à partir de l'équation Eqn.5.3 où W représente la bande passante (ici, 20 MHz) et $EbNo$ le rapport signal à bruit en Watt.

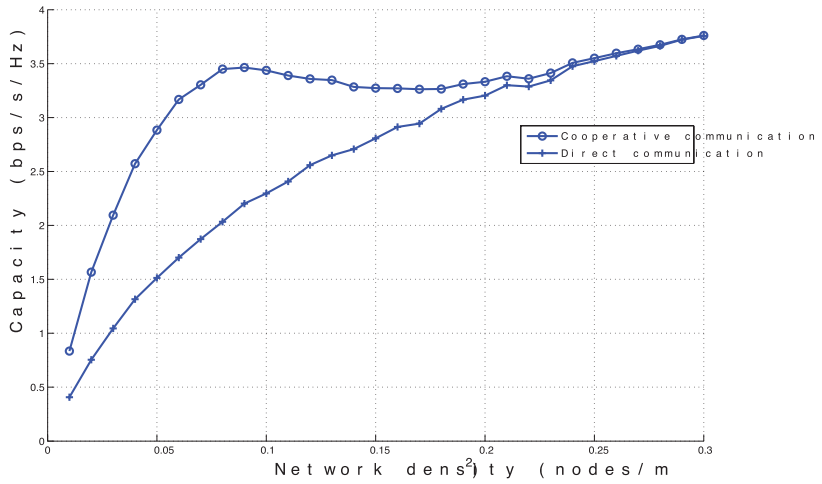

FIG. 5.6 – Capacité réseau en fonction de la densité du réseau

$$C = W * \log(1 + EbNo) \tag{5.3}$$

Lorsque la densité est de 0.08 $noeuds/m^2$, cette capacité est maximale. La capacité coopérative ainsi obtenue est bien au-delà des capacités réelles des capteurs mais permet de nous éclairer sur les possibles performances de notre solution [28].

La figure Fig.5.7 met en avant le nombre de paquets échangés entre les noeuds du réseau en incluant les retransmissions. Nous présenterons ces résultats dans trois scénarios différents.

- Les noeuds n'utilisent que des communications directes.
- Les noeuds n'utilisent que des communications coopératives.
- Les noeuds utilisent des communications directes dans un environnement parfait (pas de perte, pas de retransmission).

Dans une topologie présentant une faible densité, les communications directes utilisent beaucoup de bande passante. Le taux de pertes étant plus important, le nombre de retransmissions est lui aussi plus élevé. Lorsque la densité est de 0,05 $noeuds/m^2$, le nombre de transmissions directes est supérieur de 250% au nombre de transmissions utilisant la communication coopérative. Lors de l'utilisation des transmissions directes, le nombre de paquets émis diminue à partir de 0,05 $noeuds/m^2$ jusqu'à atteindre le niveau d'un environnement parfait lorsque la densité est de 0,15 $noeuds/m^2$. Lorsque la densité est importante, les transmissions entre les noeuds sont de meilleurs qualité. Cela explique la diminution du nombre de paquets transmis.

Nos simulations ont montré que la solution WSC-MAC permet d'améliorer le pourcentage de paquets reçus et la fiabilité du réseau lorsque la densité du réseau est faible

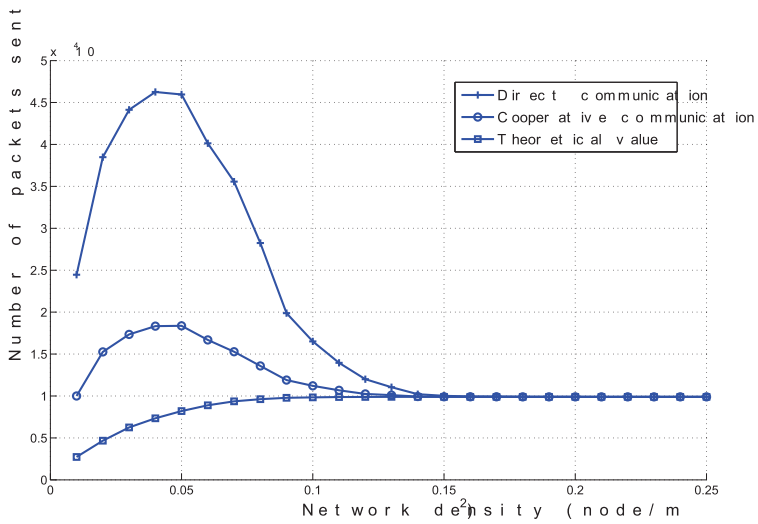

FIG. 5.7 – Comparaison du nombre de paquets envoyés et de la valeur théorique en fonction de la densité du réseau

(inférieur à 0.10 $noeuds/m^2$). Le trafic d'acquittement est également réduit. La capacité réseau utilisant les communications coopératives est bien supérieure aux capacités réels des équipements. Cela nous amène à penser que des travaux sont nécessaires dans le domaine des modulations adaptatives dédiées aux réseaux de capteurs.

5.4 Conclusion

Nous avons présenté dans cette section un mécanisme de communication coopérative adapté aux réseaux de capteurs [63]. Ce mécanisme est basé sur un identifiant de groupe qui permet de déterminer les relais potentiels parmi les voisins d'un noeud. Un mécanisme permettant de sélectionner uniquement les noeuds améliorant la communication a été ajouté. Ce mécanisme est basé sur les informations fournies par les équipements utilisant la norme 802.15.4 lorsqu'un paquet est reçu. Le mécanisme de décision peut être amélioré, notamment en ce qui concerne l'identifiant de groupe. Dans notre solution, le processus de détermination de l'identifiant de groupe est lourd et nécessite des échanges de messages lors de la phase d'autoconfiguration. Il serait nécessaire de développer un mécanisme plus performant et moins consommateur de bande passante et d'énergie.

L'apport de la coopération permet d'améliorer la fiabilité des communications dans un réseau de capteurs. L'utilisation de l'intensité du signal reçu (*RSSI*) permet d'optimiser le fonctionnement du réseau. Cette information pourrait aussi être utilisée par

d'autres protocoles pour améliorer leurs performances. On peut notamment penser aux protocoles de routage de la couche réseau.

Chapitre 6

MAODV-SIM

L'utilisation des communications coopératives, bien qu'apportant des résultats très intéressant nécessite d'avoir à dispositions des capteurs ayant une interface radio relativement évoluée. Il faut pouvoir gérer les synchronisations entre les envois et entre les différents capteurs. C'est pourquoi nous présentons maintenant une solution se basant sur une couche MAC classique qui n'implémente pas les mécanismes de coopération présentés précedemment.

La plupart des protocoles de routage utilise le nombre de sauts comme métrique pour déterminer la meilleure des routes. Cette métrique est également souvent utilisée pour les réseaux sans fils. Dans la continuité de ce que nous avons présenté précédemment, nous nous intéressons maintenant à l'utilisation de la métrique *RSSI* pour déterminer une meilleur route. Des travaux ont déja été réalisés [105] et permettent de définir la meilleure route en utilisant de cette métrique. Notre étude diffère de [105] car nous nous intéressons à la problématique de fiabilité, c'est pourquoi la notion de meilleure route est, selon nous, différente. Dans la plupart des solutions, la meilleur route est la route utilisant le moins de noeuds ou consommant le moins d'énergie ou la plus rapide. La meilleure route est donc, pour notre étude, la route la plus fiable, c'est-à-dire, celle qui assure la délivrance d'un paquet avec une probabilité maximale. Nous présenterons dans la suite de ce document, *MAODV-SIM* [64] : une solution qui nous a permis de fiabiliser les communications en utilisant la notion de route de secours et la métrique *RSSI*.

6.1 De l'intérêt du multi-chemin

Les algorithmes de routage à chemins multiples permettent, non pas d'envoyer l'information sur plusieurs chemins simultanément, mais plutôt de spéficier des routes de secours. Ces routes de secours sont alors disponibles si la première route utilisé n'est plus valide (défaillance d'un ou de plusieurs noeuds du chemin). Dans cette optique,

l'idéal serait d'avoir plusieurs routes de secours qui mènent à une même destination et qui pourraient être utilisées en cas de panne.

Pour connaitre le comportement du réseau en présence de plusieurs chemins menant à une même destination, nous avons réalisé un modèle sur Matlab [79]. Ce modèle permet de mettre en valeur l'apport des routes de secours dans le taux de délivrance des paquets.

Nous avons réalisé les simulations pour une topologie statique d'un réseau de 100 capteurs où le nombre moyen de voisins pour chaque noeud est de trois. Nous avons fait varier le taux de défaillance des noeuds sur le réseau entre 10% et 50% (*i.e.* 10%/50% des noeuds sont inopérants à différents instant t) . Les transmissions sont considérées ici comme idéales. Nous ne considérons pas ici les pertes dûes aux interférences. Seul nous intéressent les pertes dûes à la défaillance d'un ou de plusieurs noeuds. Dans cette analyse, nous avons également considéré que les différents chemins vers une même destination pouvaient utiliser une partie d'un autre chemin vers cette même destination (*i.e.*, les chemins ne sont pas obligatoirement disjoints).

Considérant :
- ω la probabilité que le paquet soit délivré,
- α la probabilité de défaillance d'un noeud,
- γ le nombre de noeuds intermédiaires entre la source et la destination,

nous obtenons l'équation suivante :

$$\omega = (1 - (1 - \alpha)^{\gamma}) \tag{6.1}$$

Considérant
- μ comme le nombre de routes disponibles vers une destination

$$z = (1 - (1 - \alpha)^{\gamma})^{\mu} \tag{6.2}$$

Le taux de réceptions des paquets dans un réseau de capteurs est alors obtenue à travers l'équation Eqn.6.2.

Les résultats obtenus sont illustrés sur les figures Fig.6.1 et Fig.6.2.

Pour 10% de défaillance, on remarque que plus de 75% des paquets pourront atteindre leur destination en présence de deux routes de secours. Avec trois routes et plus, le taux de paquets reçus avec succès atteint les 90%. Lorsque le pourcentage de défaillance est de 50%, le taux de paquets reçus avec succès est très faible : quand on a une seule route menant à destination, il est de l'ordre de 25%, ce taux s'élève à 50% si on a quatre routes de secours. Les noeuds dans un réseau de capteurs étant fragile, leur défaillance pourrait donc être paliée par l'utilisation de routes de secours.

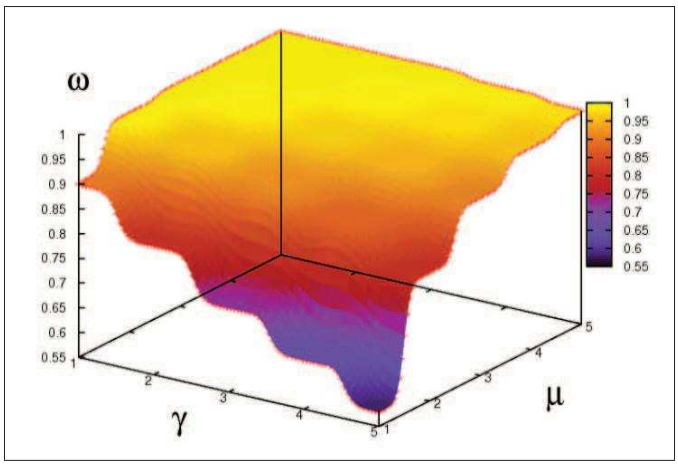

FIG. 6.1 – Taux de paquets reçus avec le multi-chemin (α=10%)

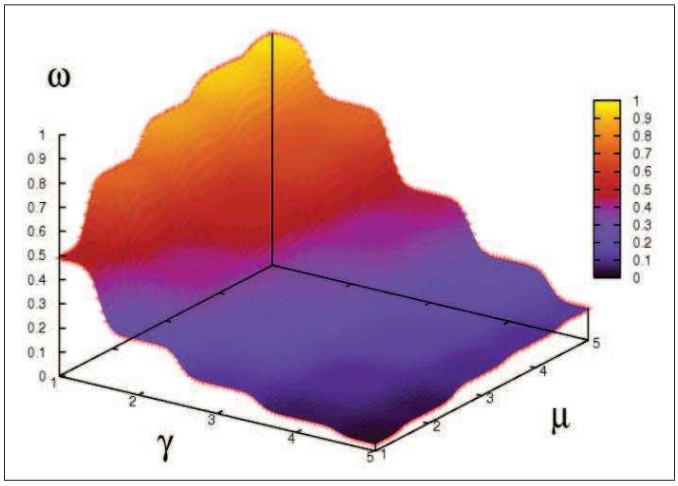

FIG. 6.2 – Taux de paquets reçus avec le multi-chemin (α=50%)

6.2 MAODV-SIM : Mutlipath-AODV based on Signal Intensity Metric

Pour améliorer la fiabilité dans un réseau de capteurs, nous avons basé notre étude sur un protocole déjà existant, AODV [75], également disponible dans le nouveau standard IEEE 802.11s [2]. Notre contribution consiste à introduire le multipath, avec *MAODV* et puis à changer de métrique de façon à privilégier les chemins qui ont l'espérance de vie la plus longue, avec *MAODV-SIM*.

6.2.1 Calcul des chemins multiples

Le recours aux chemins multiples améliore considérablement la fiabilité dans les réseaux de capteurs puisqu'il permet d'avoir immédiatement des routes de secours lorsque la route principale est inutilisable. Avec *AODV*, quand un noeud reçoit plusieurs *RREP* pour une même destination, il ne garde qu'une seule route dans sa table de routage, celle qui a le nombre de sauts minimum vers la destination. Si cette route n'est plus valide, on doit lancer une nouvelle requête afin de découvrir les nouvelles routes. Cette technique est très couteuse en ressources et en délai pour un réseau de capteurs. Il faut, en effet, redémarrer entièrement le processus de détermination de roure (*RREQ/RREP*). Cela peut être évité si on dispose de routes de secours accessibles directement en cas de panne. Si toutes les routes de secours ne sont plus utilisables, une requête pour une nouvelle route pourra être lancée.

Dans notre solution, on garde plusieurs routes vers une même destination dans la table de routage. En effet, pour chaque *RREP* reçu, on ajoute une route dans la table de routage, on ne garde pas que la meilleure route (*i.e.* celle qui possède le moins de sauts). Ainsi, quand la route principale n'est plus active, on peut utiliser une deuxième route déjà présente dans la table de routage, ce qui nous fera gagner en terme d'*overhead* et de délai. Cette technique permet l'amélioration de la fiabilité du réseau mais ce mécanisme nécessite des optimisations supplémentaires. En effet, la présence de plusieurs routes et le maintien du nombre de sauts comme métrique ne nous permettent pas de choisir la meilleure route vers une destination (*e.g.* si plusieurs routes possèdent le même nombre de sauts). L'utilisation d'une nouvelle métrique nous paraît alors indispensable.

6.2.2 Métrique d'intensité du signal

Dans les réseaux filaires classiques et dans certains réseaux sans fils, l'utilisation du nombre de sauts comme métrique est relativement courante. Dans un réseau de capteurs, le nombre de saut n'est pas forcément une métrique pertinente. En effet, l'énergie étant une ressource limitée, il est plus intéressant de réduire la quantité d'énergie nécessaire à l'envoi d'un paquet que de réduire le nombre total de noeuds relayant l'information. L'énergie est en effet directement liée à la distance séparant physiquement deux noeuds. Egalement, dans notre étude, la fiabilité est fonction du nombre de sauts mais elle est également fonction de la qualité du lien entre les noeuds. Nous cherchons à définir ce qui est, selon nous, la meilleur route : la route la plus fiable.

Dans notre solution, nous avons donc introduit la métrique d'intensité du signal reçu : *signal intensity metric*. Cette métrique est semblable à la métrique que nous avons utilisée précédemment dans le cadre de communication coopérative. Elle est basée sur la norme IEEE 802.15.4. Le standard définit deux types de mesure caractérisant une transmission : *LQI* pour *Link Quality Indication* et *ED* pour *Energy Detection*

Distance (m)	Signaux Reçus au capteur A		Signaux Reçus au capteur B	
	RSSI (dBm)	*LQI*	RSSI(dBm)	*LQI*
1,5	-56,38	107,13	-55,63	107,1
3	-59,3	107,62	-58,19	107,3
4,5	-60,52	107,41	-61,19	107,26
6	-64,18	107,17	-63,97	107,05
7,5	-69,49	108,04	-69,08	107,69
9	-68,71	107,83	-69,04	107,44
10,5	-69,55	107,76	-69,51	107,44
12	-69,29	107,04	-69,54	106,88
13,5	-69,44	107,65	-59,89	107,08
15	-69,97	107,75	-69,79	107,06
16,5	-79,71	107,48	-79,18	107,18
18	-76,64	107,21	-76,91	106,91
19,5	-79,83	107,07	-69,99	106,67
21	-79,04	106,88	-78,81	106,67
22,5	-77,28	107,58	-77	107,26

TAB. 6.1 – RSSI et *LQI* en fonction de la distance

également appelé *RSSI Receive Signal Strength Indicator*. Dans une précédente étude [82] nous avions analysé la qualité des signaux reçus etre deux capteurs(capteur A et B) Micaz [107] en fonction de la distance. Ces capteurs utilisent des chipset radio compatible 802.15.4. Le but de cette étude était d'analyser les différences entre le *RSSI* et le *LQI* [1] mais également de quantifier la diminution du signal en fonction de la distance.

Les résultats obtenues sont présentés dans le tableau 6.1. On y voit clairement une symétrie dans les niveaux d'énergie des signaux reçus par chacun des noeuds. On peut également remarquer que la valeur de *LQI* et *RSSI* varient en fonction de la distance. La pertinence de l'utilisation de l'une ou l'autre des métriques proposées par la norme n'est cependant pas visible. On remarque néammoins que le chipset radio CC2420[] de ce type de capteur fourni une valeur de *RSSI* correspondant réellement aux signaux reçus. La valeur de *LQI* est, elle, calculé à partir des messages reçus et peut-être considérée comme étant une mesure du*chip error rate*. Pour un chipset CC2420, la caractérisation d'un signal varie de 110(qualité maximale) à 50(qualité minimale). Certaines études traitent de ce thème en profondeur [88, 81] et établissent les caractéristiques de chacune des métriques.

La qualité de la transmission reçue nous permet de caractériser le lien. Nous considérons que, plus l'intensité du signal reçu est importante, plus la liaison entre les deux noeuds est fiable. Les valeurs *LQI* et *ED* sont inversement proportionnelle à

la distance séparant deux noeuds. Il existe d'autres facteurs environnementaux pouvant perturber une transmission : arbres, bâtiments ... Dans le cas d'une visibilité directe claire (*clear line-of-sight*), le principal paramètre faisant varier le *LQI* et le *ED* est la distance séparant la source de la destination. Dans le cas d'un réseau mobile, l'utilisation de cette métrique peut alors être très pertinente. Lorsque deux noeuds sont éloignés, leurs valeurs de *LQI* et *ED* sont faibles et peuvent témoigner d'une future rupture de lien entre ces deux noeuds. Le protocole de routage doit donc prendre en compte cela afin d'évaluer les différentes routes.

Notre protocole ne permet pas de déterminer le plus court chemin mais le chemin le plus fiable. Le choix de la meilleure route est fait par la source après avoir reçu des réponses de routes (*RREP*). La *meilleur* route est défini comme étant la route possèdant la plus haute valeur d'intensité du signal reçu dans la table de routage.

6.2.3 Détails protocolaires de MAODV-SIM

Nous avons défini deux algorithmes permettant de définir la route la plus fiable. Le premier est exécuté par le noeud source afin de déterminer la meilleure route (*cf.* Alg.4). Le deuxième est utilisé par chaque noeud recevant une réponse à une requete de route (*cf.* Alg.5).

L'ajout d'un champ dans l'en-tête de la réponse *RREQ* est nécessaire. Ce champ, *Signal Intensity of the Received Route (SIRR)* (*cf.* Fig.6.3), permet de stocker la valeur du signal reçu du chemin.
C'est ce champ qui permettra au noeud source de caractériser le chemin correspondant au paquet *RREQ* reçu. Il est tout d'abord positionné à une valeur prédéfinie par l'initiateur de la réponse. Celui-ci envoie, comme prévu par le protocole initial, cette réponse vers la source, . Lorsqu'un noeud la reçoit, il va comparer la valeur qui est stockée dans le champs *SIRR* du paquet avec l'intensité du signal reçu de ce paquet (*SIRM*). La valeur *SIRR* de la réponse transférée au noeud suivant est alors la plus faible. De cette façon, lorsque le paquet aura parcouru le chemin complet jusqu'à la source, le champ *SIRR* contiendra l'intensité du signal la plus faible enregistrée sur l'ensemble du trajet.

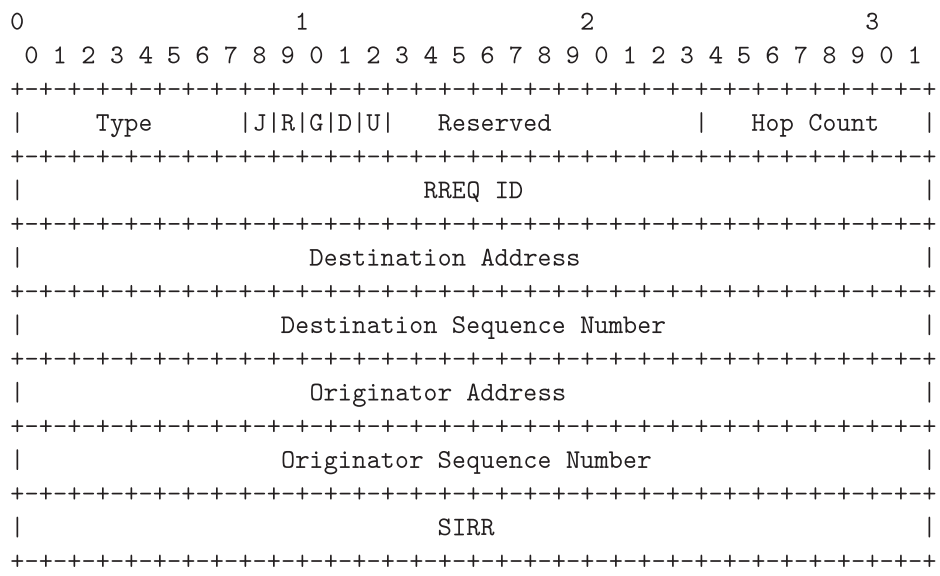

FIG. 6.3 – En-tête modifié de MAODV-SIM

Input: SIRM,SIRR
Output: Route fiable

if *(Noeud recoit une réponse Route Reply)* **then**
  if *(SIRM ≤ SIRR)* **then**
    SIRR = SIRM;
  end
  if *(Je suis l'initiateur de la demande de route)* **then**
    Ajout de la route dans la table de routage;
  else
    Transférer la réponse au noeud suivant ;
  end
end

Algorithm 4: Transfert de réponse *Route Reply Forwarding*

Lorsque la réponse est reçue par la source, la route est ajoutée à la table de routage existante, même dans le cas où une route serait déjà présente [*cf.* Alg.4]. Pour chacune de ces routes, nous avons une valeur différente d'intensité du signal : *Signal Intensity Route (SIRoute)*. Lorsque la source enverra un message, elle utilisera la meilleure route, c'est-à-dire celle ayant le SIRoute le plus élevé.

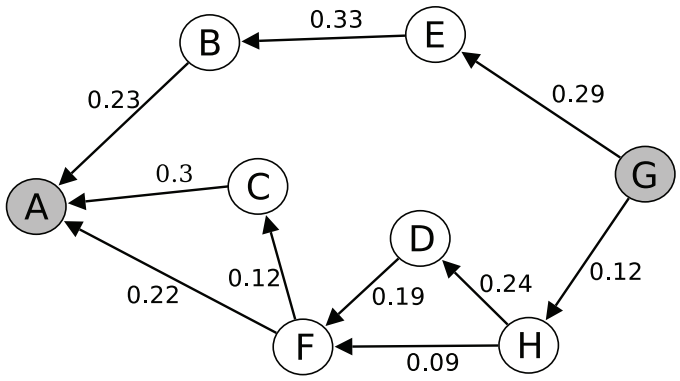

FIG. 6.4 – Exemple : définition de route la plus fiable

Input: Paquet à envoyer
Output: Route

if *(Paquet à envoyer)* **then**
  lire l'adresse de destination du paquet ;
  if *(destination existe dans la table de routage)* **then**
    Choisir la route avec le SIRoute le plus élevé;
    Positionner l'indicateur d'utilisation de route;
    Retourner la route choisie;
  else
    Envoyer une requête de route;
  end
end

Algorithm 5: Choix d'une route dans la table de routage

La table de routage possède un indicateur d'utilisation de route qui permet de savoir si une route a été utilisée. Cet indicateur doit être positionné lors de l'envoi d'un paquet. En effet, si la route n'est plus utilisable, le protocole doit être capable de sélectionner une route parmi les routes de secours disponibles. Il va choisir la route ayant la valeur de SIRoute la plus élevé et dont l'indicateur d'utilisation de route n'est pas positionné. La meilleure route sera ainsi différente de celle utilisée précédemment mais restera la meilleure des routes de secours disponibles.

La figure 6.4 présente un réseau de capteurs dans lequel le noeud A souhaite envoyer un paquet au noeud G. Pour cela, il envoie une requête *RREQ* à ses voisins jusqu'à ce que celle-ci atteigne sa destination. Les réponses *RREP* sont symbolisées par les flèches. Le poids de chaque liaison représente l'intensité du signal reçu *SIRM* (*e.g.* l'intensite du signal par E depuis G est 0,29). Le noeud A va alors recevoir plusieurs réponses *RREP* correspondant à plusieurs chemins possibles. Notre algorithme nous

permet alors de définir la route la plus fiable (dans ce cas, la route A-B-E-G). En d'autres termes, notre solution permet de définir la route ayant la plus haute valeur minimum de *RSSI* de chaque chemin.

6.3 Analyse et performance de MAODV-SIM

Afin d'évaluer notre protocole, nous l'avons implémenté et testé en utilisant le simulateur NS-2. Nous avons comparé notre solution avec le protocole *AODV* implémenté sous *AODV*. Le but de la démarche n'est pas seulement de prouver que notre solution améliore les performances d'*AODV*, mais qu'elle devrait permettre d'améliorer les performances des algorithmes de routages réactifs.

Nous allons donc prouver que la métrique d'intensité du signal reçu et l'utilisation de routes de secours permettent d'augmenter la fiabilité du réseau. Pour quantifier la fiabilité, nous considérons le taux de paquets reçus. Nous pouvons mesurer et comparer la quantité des paquets perdus lorsque notre solution est utilisée et lorsqu'elle ne l'est pas. Plus ce taux est élevé, plus notre solution est considérée comme fiable.
Nous avons également voulu évaluer notre solution par rapport à sa réaction aux changements de topologies.

Un noeud est un équipement faible qui peut être inopérant pour différentes raisons : destruction physique, réserves énergétiques épuisées, défaillance logicielle ... La topologie peut donc être modifiée rapidement et une route devenir inutilisable. Dans ce genre de situation, le réseau ne doit pas être perturbé et les données doivent, autant que faire se peut, atteindre leurs destinations. Afin de mettre en avant la faiblesse des noeuds et son impact sur le protocole de routage, nous avons simulé cette faiblesse par l'extinction volontaire de noeuds (définie aléatoirement) à un instant précis. Cette défaillance "simulée" des noeuds du réseaux peut entrainer la coupure d'une route et donc empecher le transfert d'un paquet entre source et destination.

La théorie nous a montré que les routes de secours permettent d'augmenter le taux de paquets reçus. Notre solution implémente un mécanisme de routes de secours que nous allons donc évaluer. Nous comparerons notre solution au protocole *AODV* de base ainsi qu'à l'évolution d'*AODV* que nous avons définie : *MAODV* (route de secours avec métrique du nombre de saut).

Pour des raisons de clarté, nous avons résumé les paramètres de simulations dans la table Tab.6.2.

Fiabilité Dans un premier temps, nous nous intéressons donc au taux de paquets reçus en fonction du nombre de noeuds défaillants (*cf.* Fig.6.5). Les performances de *MAODV* et *MAODV-SIM* sont très proches. La moyenne des taux de paquets reçus en utilisant *MAODV-SIM* est supérieure à *MAODV* de 2,5%. L'amélioration vis à vis de *AODV* est, elle, beaucoup plus importante, la moyenne des taux de pa-

Paramètres	Valeur
Paramètre de Simulations	
Nombre de noeuds	50
Nombre de noeuds défectueux	0-20
Densité de noeuds (noeuds/m^2)	0,0002
Nombre d'itérations	20
Vitesse de déplacement (m/s)	0-10
Paramètres MAC/PHY	
Portée de transmission (mètres)	100
Modèle de propagation	Two-ray ground
Modèle Physique	802.11b
Trafic	Constant Bit Rate
Sensibilité à la reception (dbm)	-90
taille du paquet de données (bits)	625

TAB. 6.2 – Paramètres de Simulation

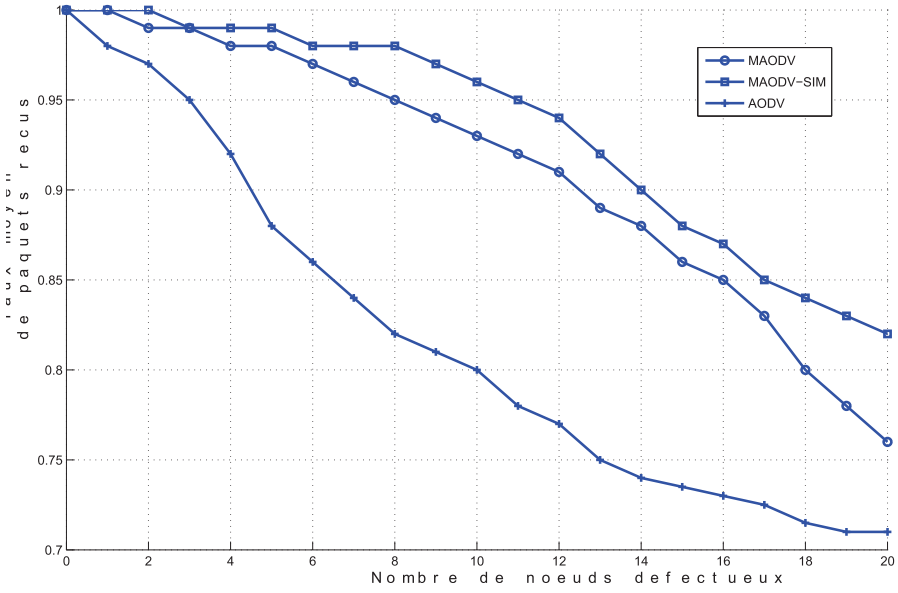

FIG. 6.5 – Taux de paquets reçus en fonction du nombre de noeuds défectueux

quets reçus est supérieure de 15% avec *MAODV-SIM*. Ce résultat corrobore la théorie présentée précédemment qui tend à prouver que l'utilisation de routes de secours per-

met d'augmenter le nombre de paquets reçus. La métrique de niveau d'énergie permet ici d'améliorer les performances par rapport à *MAODV*. Dans le cas de l'utilisation de cette métrique, les routes les plus fiables seront privilégiées et le taux de paquets reçus sera donc supérieur (*MAODV* définit la meilleur route selon le nombre de saut qui la compose). Les routes de secours ont permis à *MAODV* et *MAODV-SIM* de trouver une nouvelle route lorsque le paquet a été perdu.

Notons également que notre solution réagit mieux aux défaillances du réseau. Les taux de réceptions sont relativement constants lorsque moins de 20% des noeuds sont défaillants. Dans ce cas, plus de 95% des paquets sont reçus. Lorsque 30% des noeuds du réseau sont défaillants, *MAODV-SIM* permet la réception de 90% des paquets. *AODV* ne permet, lui, que la réception de moins de 75% des paquets émis. Dans le cas d'un réseau "perturbé", l'utilisation de routes de secours et une métrique plus appropriée permettent donc d'assurer une meilleure délivrance des paquets et donc une meilleure fiabilité.

Délai Le délai de bout-en-bout représente le délai moyen d'acheminement d'un paquet de la source à la destination. Il prend notamment en compte le temps nécessaire à la détermination d'une nouvelle route si la première s'est avérée inutilisable. Dans le cas de *MAODV-SIM*, on conserve plusieurs routes différentes vers une même destination. Il n'est donc pas nécessaire de redéfinir une nouvelle route si la première est défaillante. Dans le cas d'*AODV*, les noeuds défaillants invalident les routes et cela nécessite la découverte d'une nouvelle route, entraînant alors une augmentation du délai de bout-en-bout (*cf.* Fig.6.6).

Mobilité Dans le cas d'un réseau mobile, l'utilisation de routes de secours et de la nouvelle métrique apporte également une amélioration. En moyenne, on observe un taux de réception des paquets supérieur de 3% avec *MAODV-SIM* (*cf.* Fig.6.7) Cela s'explique essentiellement par l'utilisation de la métrique. En effet, cette dernière privilégie les liens les plus fiables entre chacun des noeuds d'un chemin. Lorsque deux noeuds sont éloignés, l'intensité du signal reçu décroit et le lien peut alors être rompu entre eux si l'un des deux noeuds se déplace et s'éloigne de l'autre. La mobilité est donc un facteur déterminant dans la réception des paquets.

Nous avons donc déplacé aléatoirement les noeuds à travers l'environnement en faisant varier leur vitesse de déplacement. Dans cette simulation, nous n'avons pas simulé la défaillance de noeuds. On observe alors que, si la vitesse est supérieure à 5 m/s, le taux de paquets reçus avec *MAODV-SIM* est supérieur de 4% au taux de paquets reçus avec *AODV*. Dans le cas d'un réseau de capteurs, les noeuds sont en général statiques et leur vitesse de déplacement est donc réduite. Néanmoins, ces résultats tendent à démontrer que notre solution permettrait d'améliorer les performances réseaux dans d'autres scénarios et d'autres domaines. s

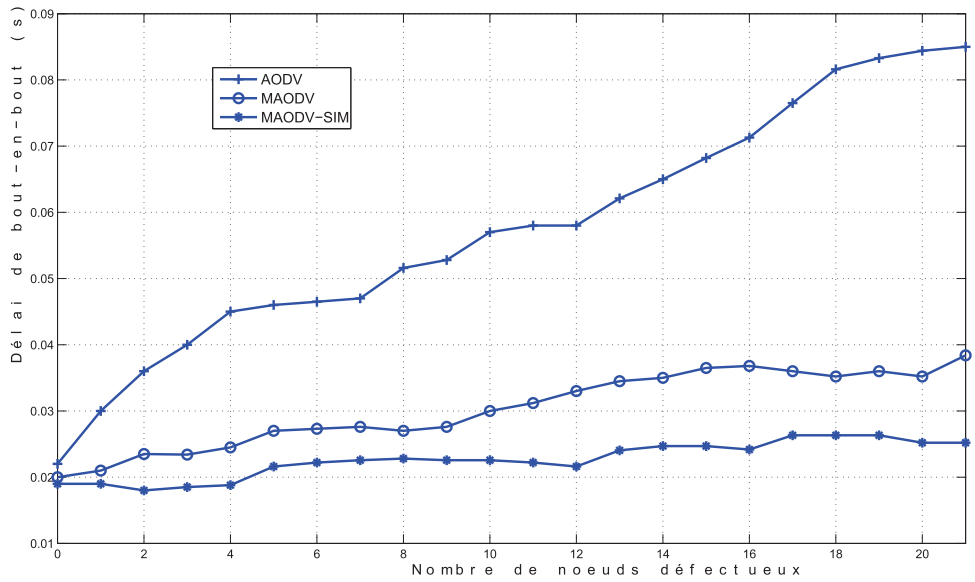

FIG. 6.6 – délai de bout-en-bout en fonction du nombre de noeuds défectueux

Débit Comme pour la mobilité, le débit disponible n'est pas une caractéristique fondamentale d'un réseau de capteurs. Dans la plupart des scénarios actuels, le volume d'informations générés et transmis par le capteur est très faible, de l'ordre de quelques Ko par seconde. Le débit n'est donc pas déterminant dans le déploiement d'application capteurs. Néanmoins, nos résultats permettent de montrer que *MAODV-SIM* résiste mieux aux défaillances de noeuds et apporte une amélioration moyenne de 10% sur le taux de paquets reçus (*cf.* Fig.6.8). Son utilisation pourrait donc être envisagée dans d'autres types d'applications.

6.4 Conclusion

La solution MAOD-SIM [64] présentée dans cette section est basé sur l'utilisation de route de secours et sur l'utilisation permettant d'identifier avec plus de précision les routes dites "fiables".

Les résultats présentés précedemment ont mis en avant son efficacité. Nous avons en particulier focalisé notre travail sur la fragilité des réseaux de capteurs. Cette fragilité des capteurs peut entrainer notamment l'invalidité de routes définies par le protocole de routage.

Bien que le débit ne soit pas, selon notre avis, un paramètre pertinent de la fiabilité d'un réseau, on observe que nos mécanismes permettent d'améliorer sensiblement le

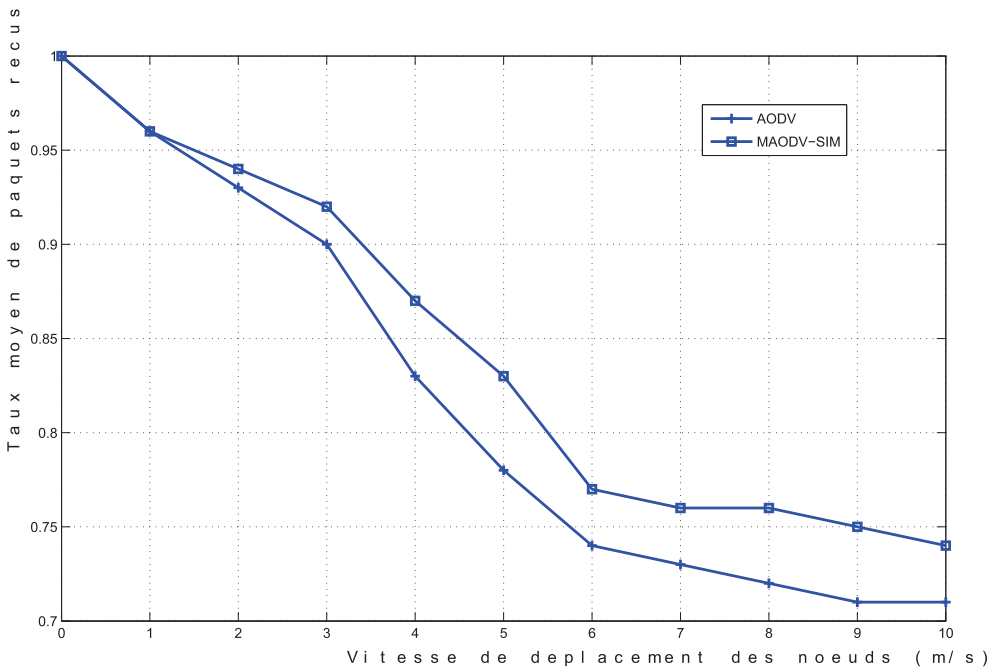

FIG. 6.7 – taux de paquets reçus en fonction de la vitesse de déplacement des noeuds

débit. On observe notamment que le débit est moins impacté par la défection des noeuds.

Les paramètres plus représentatifs de la fiabilité se voit également améliorés par notre solution.

Le délai de bout-en-bout, contrairement au débit, peut être considéré comme un paramètre pertinent de la fiabilité d'un réseau. En effet, la transmission de l'information dans un réseau de capteur est capitale. Cette donnée doit arriver au destinataire sans pertes d'intégrité mais également arrivé "dans les temps". Une donnée reçue trop tard et une donnée inutile. Les mécanismes que nous avons mis en place montre que le délai de bout-en-bout moyen observé est plus stable avec notre solution.

Enfin, on observe un taux de réception de paquets plus élevé lorsque l'on utilise notre solution, et donc, selon nos critères, une fiabilité accrue. Il est à noté que bien que l'utilisation de la nouvelle métrique et de la route de secours permet une fiabilité optimale, c'est l'utilisation seule de la route de secours qui permet la plus importante augmentation de paquets reçus.

Dans cette étude, nous avons utilisé le protocole AODV pour développer et implémenter notre solution. Les principes de route de secours et d'identification de la route la plus fiable par l'utilisation de nouvelles métriques ne sont pas propre à AODV.

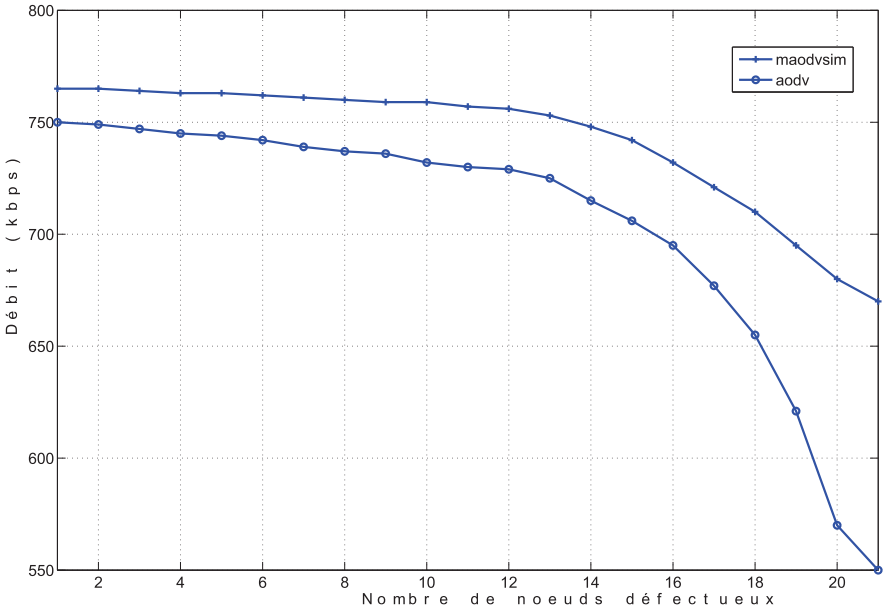

FIG. 6.8 – Débit moyen en fonction du nombre de noeuds défectueux

Nous pensons que ces mécanismes devraient pouvoir être adapté à d'autres protocoles de routages et ainsi permettre d'améliorer la fiabilité des réseaux de capteurs ou des réseaux de type Manet.

Conclusion

Dans ce chapitre, nous avons étudié la fiabilité des réseaux de capteurs et proposées des solutions permettant de l'améliorer : les communications coopératives situées au niveau de la couche liaison du modèle OSI et la coopération au niveau de la couche réseaux du modèle OSI.

La coopération au niveau de la couche liaison est basée sur l'envoi simultané d'une même information depuis deux entités différentes vers une seule et même entité. La coopération au niveau de la couche réseaux est basée sur l'utilisation de routes de secours pour réduire le trafic de signalisation et améliorer la fiabilité des transmissions. Dans ces deux types de coopération, nous avons utilisé une métrique spécifique permettant de quantifier les liens entre les noeuds. Cette métrique est basée sur la norme 802.15.4. Cette dernière définit deux mesures qui sont fournies par la couche physique des équipements normalisés 802.15.4. L'utilisation de cette métrique permet d'optimiser la coopération en privilégiant les liaisons de meilleures fiabilités.

Les performances observées montrent une plus grande fiabilité des transmissions et une réduction du trafic. Dans le cas d'une application médicale, ces améliorations apportent un intéret certain. En effet, une application médicale est, par définition, critique et la fiabilité est donc une composante importante. Dans nos études, nous avons démontré que le taux de paquets reçus, notre quantifieur de fiabilité, est supérieur grâce à nos solutions.

La consommation d'énergie dans les réseaux de capteurs est une problématique essentielle. Elle l'est d'avantage dans les applications médicales. Les applications de télésurveillance utilisant des équipements doivent être énergétiquement autonomes. Un équipement doit avoir une durée de vie maximale. Cela passe par la définition de mécanisme d'économie d'énergie. La réduction du trafic permet notamment de réduire l'énergie utilisée pour la transmission et la réception d'informations. La réduction du trafic permet également d'améliorer la fiabilité de l'application. Un canal libre permet un accès au canal plus rapide et permet également de réduire le nombre de collisions.

Chapitre 7

Conclusion Générale et Perspectives

Conclusion

Les objectifs que nous avions fixés au début de la thèse était de réaliser une plate-forme de communication dédiée à l'assistance aux personnes. Cette plate-forme a été dévéloppée, déployée et testée dans un environnement à fortes contraintes. Cette intégration nous a alors permis de réaliser diverses expérimentations en environnement réels. Nos observations nous ont permis d'obtenir des résultats ainsi que des analyses comportementales d'usagers d'équipements de communications mobiles.

Les résultats que nous avons obtenus sont très encourageant et nous ont permis de démontrer la viabilité d'une telle plate-forme dans des environnements clos. Nous avons notamment pu observer et quantifier les faiblesses de la technologie Bluetooth et proposer des solutions pouvant palier à ses faiblesses. Cette plate-forme nous a alors permis tout au long de cette thèse de proposer des solutions en adéquation avec nos observations.

Dans le chapitre 3, nous avons notamment proposé une solution de réduction de l'utilisation de la bande passante dans un réseau de capteurs. La principale contribution de cette solution nous a permis de valider le postulat suivant : la compression de données et l'envoi de ces données compressées consomment moins d'énergie que l'envoi direct des données brutes. Le souhait que nous avions à l'initial était d'être le plus proche possible de la réalité. C'est pourquoi nous avons utilisé des outils permettant de connaitre avec une très grande précision la consommation d'énergie des fonctions utilisées sur les capteurs. Les résultats obtenus ont démontré une économie d'énergie significative mais également une réduction de la bande passante.

Rapidement, les discussions avec les différents partenaires du projet de la plate-forme ont fait ressortir une composante sécurité qu'il était important de ne pas négliger.

La plate-forme étant en phase de développement et non d'exploitation, nous n'avons pas implémenté de solutions de sécurité. Mais nous avons tout de même travaillé à l'élaboration d'une solution de sécurité basée sur une telle architecture. Nous avons basé notre travail sur une architecture dans laquelle les réseaux de capteurs sont liés à un réseau IP via des passerelles intelligentes (à l'image des points d'accès de notre plate-forme). Notre solution permet la création d'un tunnel de bout-en-bout entre un nœud et un terminal de surveillance situé sur le réseau IP. Le tunnel est crée via une passerelle entre les deux différents réseaux. La passerelle étant finalement en position de *man-in-the-middle*, il a été nécessaire de définir une solution de confiance envers celle-ci. Nous avons défini le concept de confiance partielle et totale qui nous a permis de proposer deux variantes de notre solution pouvant s'adapter aux ressources disponibles ainsi qu'à la politique de sécurité désirée. Les résultats ont notamment démontrés de meilleures performances que les solutions existantes, en particulier sur les temps des opérations de chiffrement et de déchiffrement.

Dans la partie suivante, nous nous sommes intéressés à la problématique de fiabilité des réseaux de capteurs.

Différents travaux ont démontré que les réseaux de capteurs étaient des réseaux fragiles et que des disfonctionnements pouvaient subvenir. La perte de connectivité, de paquets, les délais de bout-en-bout rallongés en cas de route défaillante font partie des problèmes que nous avons tentés de résoudre.

Dans le chapitre 5, nous avons proposé une solution permettant d'assurer une meilleure délivrance des paquets par le biais de communications coopératives. A l'instar des antennes MIMO qui transmettent depuis un équipement possédant plusieurs antennes, le même message simultanément, nous avons proposé des solutions où les antennes sont distribuées sur différents capteurs. Dans le protocole que nous avons proposé, un message est envoyé par deux capteurs différents vers un troisième capteur, destinataire du message. Pour ce protocole, nous nous sommes en particulier concentrés sur le développement d'une solution efficace de définition du nœud relais. En effet, il est nécessaire de définir un nœud relais ayant la charge d'envoyer lui aussi le paquet vers la destination. Les différents travaux consultés utilisaient des mécanismes complexes permettant de définir le nœud relais et nécessitant de nombreux envoi de message et donc inadaptés à un réseau aux ressources limitées. La plupart des autres solutions existantes, font abstraction de cette démarche et considère le nœud relais comme étant défini sans préciser comment. Notre solution permet de définir de façon automatique le nœud devant relayer l'information. L'utilisation de ce mécanisme et d'une métrique plus adaptée (RSSI) nous a permis de définir le protocole WSC-MAC. Par le biais de simulation, ce protocole a démontré ses performances, notamment en ce qui concerne les taux de pertes de paquets.

Enfin, dans le dernier chapitre, nous nous sommes intéressés aux problèmes de connectivité, de fiabilité et de perte de paquets. Nous avons pour cela, développé le

concept de route de secours qui permet à un nœud de conserver plusieurs chemin vers une même destination. Ce concept a alors été utilisé au sein d'un protocole issu d'AODV : MAODV-SIM. Ce protocole utilise le concept de route de secours mais utilise également une nouvelle métrique à l'instar de WSC-MAC. Cette métrique nous permet de quantifier la fiabilité des routes plutôt que leur coût énergétique. Dans ce protocole, nous avons recherché à améliorer la fiabilité. Pour cela, notre solution permet de trouver la route la plus fiable parmi les différentes routes disponibles afin de s'assurer que le paquet transmis a une probabilité maximale d'atteindre sa destination. La sélection de la route la plus fiable prend en compte chacun des sauts qui composent son chemin entre la source et la destination et permet alors de trouver la route ayant la meilleure liaison possible. Le protocole lui permet également de conserver les autres routes qu'il aurait reçues. Ainsi, si la première route est inutilisable, il peut en utiliser une autre sans avoir besoin de recommencer le mécanisme de détermination de route qui peut consommer beaucoup d'énergie. Les résultats obtenus ont montré de meilleures performances mais ont surtout permis de valider le concept de route de secours et de valider le mécanisme de sélection de la meilleure route qui peut alors être utilisée avec d'autres protocoles.

Perspectives

Au vu des résultats obtenus, plusieurs axes de recherches se dégagent. Il existe de nombreux protocoles de routage pour les réseaux de capteurs, mais la plupart n'ont comme objectifs que de réduire le délai d'acheminement des données. Dans certains domaines d'application, ce n'est pas le délai d'acheminement des données qui importe le plus mais la réception d'un maximum de données, ou de la totalité de ces données. On pense notamment aux applications dites sensibles de surveillance médicale ou de surveillance de milieu à risque (e.g. centrale nucléaire).

Les problématiques de fiabilité et de connectivité sont des axes de recherches qui permettraient de répondre à certaines exigences dépendant de l'application. L'utilisation de métrique différente des métriques actuelles fait parti des évolutions possibles. On peut notamment penser à la métrique d'intensité du signal reçu ou au nombre de voisin d'un nœud. Ces métriques existent et sont d'ors et déjà utilisées, mais il reste à les utiliser pour déterminer des solutions fiables d'acheminement des données. Ces solutions passeront probablement par l'utilisation d'une solution d'adressage IPv6. L'émergence d'IPv6 sur les réseaux de capteurs via le groupe de travail 6LowPAN de l'IETF préfigure de l'avenir des réseaux de capteurs à savoir une interconnexion plus importante entre le réseau IP classique (e.g internet) et les réseaux de capteurs. L'apport d'IPv6 amène également un profond changement dans la philosophie des réseaux de capteurs. Classiquement, les réseaux de capteurs fonctionnent sur un principe de communication many-to-one. Plusieurs capteurs transmettent une information vers un même sink, ce sink identifie la provenance de l'information de façon globale. Dans

certains cas, le capteur ne possède pas d'identifiant propre. L'utilisation d'IPv6 permettra alors de passer à une communication one-to-one : ou il sera possible alors de connaître avec précision la provenance d'une information. Là où la circulation de l'information était unidirectionnelle : du capteur vers le sink. IPv6 permettra la définition de communication du sink vers le capteur. On passe alors vers des communications bidirectionnelles où le sink est capable d'interroger précisément un capteur.

Cette « jonction » de ces deux types de réseaux technologiquement différent et disposant de ressources très différentes, impose le développement de nouveaux protocoles, qu'il soit réseau ou transport qui autorise l'acheminement des données de façon transparente sur ces réseaux. Ces solutions, dans un souci d'efficacité, devront pouvoir s'adapter aux différents supports physiques et aux technologies sur lesquels elles évoluent sans pour autant nécessité l'utilisation de passerelle entre ces technologies.

Sur un aspect sécurité, la présence de nombreuse solutions permet d'établir une politique de sécurité cohérente mais nécessite d'être adapté à la technologie sous-jacente. En effet, on ne peut utiliser la même solution de sécurité pour un capteur utilisant un microprocesseur cadencé à 8 MHz et un ordinateur personnel utilisant un bi-processeur cadencé à 3 GHz. Si les protocoles de routages et de transport doivent s'adapter à la technologie, il en sera de même pour les protocoles de sécurité.

L'utilisation de solution de techniques de cross-layer est également un axe de recherche envisagé. Dans notre étude, nous avons notamment présenté un algorithme de réduction de bande passante utilisant un algorithme de compression de données. Ce type de solution préfigure des travaux éventuels, à savoir une analyse sémantique des données dans le mécanisme d'acheminement des données, à l'image de la qualité de service qui définit différentes classes de données avec différentes priorités. Ces mécanismes doivent être capables de décider de la politique de transmission de l'information en fonction du contenu des données (sémantique) et non plus de leur type (syntaxe).

Un axe de recherche que nous avons abordé et qui nécessiterait de futurs développements concerne les communications coopératives pour les réseaux sans fils. Il existe de nombreux travaux faisant état des bénéfices de la coopération pour les réseaux sans fils. Nous avons notamment démontré l'apport des communications coopératives en ce qui concerne la fiabilité des communications par le biais de mécanisme adapté aux réseaux de capteurs. Néanmoins, la plupart des travaux que nous avons étudiés ne prenne en compte que l'aspect communications coopératives en lui-même et non la mise en œuvre de cette coopération. La détermination du ou des nœuds relais participant à la communication est un point rarement abordé dans la littérature. Dans la plupart des travaux, le ou les nœuds relais sont désignés et considérés comme le ou les meilleurs des relais possibles. C'est cette « désignation » qui doit bénéficier de mécanismes spécifiques. C'est encore plus vrai dans le cas de réseau de capteurs où les ressources sont limitées. Les mécanismes doivent déterminer le meilleur nœud relais

possible et le faire en un minimum de communications afin de ne pas dépenser les ressources inutilement.

Bibliographie

[1] CC2420 Datasheet : 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. March 2007.

[2] IEEE 802.11s - Mesh Networking Task Group. http ://grouper.ieee.org/groups/802/11/Reports/tgs_update.htm.

[3] IEEE 802.18 - Radio Regulatory Technical Advisory Group (RR-TAG). http ://www.ieee802.org/18/.

[4] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8) :102–114, Aug 2002.

[5] W. Al-Khateeb, H. Moinudeen, and A. N. M. AbdullahShnayder. Scalability analysis of bluetooth technology. In *Proc. of Malaysian Science and Technology Conference (MSTC'03)*, 2003.

[6] S.M. Alamouti. A simple transmit diversity technique for wireless communications. *IEEE J. Sel. Areas in Comm.*, 16(8) :1451–1458, Oct 1998.

[7] Geoff W. Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. Fidelity and yield in a volcano monitoring sensor network. In *OSDI '06 : Proceedings of the 7th symposium on Operating systems design and implementation*, pages 381–396, Berkeley, CA, USA, 2006. USENIX Association.

[8] J. Andersen, B. Lo, and Guang-Zhong Yang. Experimental platform for usability testing of secure medical sensor network protocols. pages 179–182, June 2008.

[9] Junaid Ansari, Janne Riihijarvi, Petri Mahonen, and Jussi Haapola. Implementation and performance evaluation of nanomac ; a low-power mac solution for high density wireless sensor networks. *Int. J. Sen. Netw.*, 2(5/6) :341–349, 2007.

[10] Avispa : Automated Validation of Internet Security Protocols and Applications. http ://avispa-project.org.

[11] Avrora, The AVR Simulation and Analysis Framework. http ://compilers.cs.ucla.edu/avrora/.

[12] Guillermo Barrenetxea, François Ingelrest, Gunnar Schaefer, and Martin Vetterli. The hitchhiker's guide to successful wireless sensor network deployments. In *SenSys '08 : Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 43–56, New York, NY, USA, 2008. ACM.

[13] J.P. Benson, U. Roedig, A. Barroso, and C.J. Sreenan. On the effects of aggregation on reliability in sensor networks. In *Proc. of the 65th IEEE Vehicular Technology Conference (VTC'07)*, pages 145–149, 2007.

[14] E. Biagioni and S. H. Chen. A reliability layer for ad-hoc wireless sensor network routing. In *Proc. of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 9*, 2004.

[15] Bluetooth Access Server. http ://www.bluegiga.com.

[16] D.G. Brennan. Linear diversity combining techniques. *In Proc. of the Institute of Radio Engineers*, 47(6) :1075–1102, June 1959.

[17] S. Brown and C.J. Sreenan. A study on data aggregation and reliability in managing wireless sensor networks. In *Proc. of the IEEE International Conference on Mobile Adhoc and Sensor Systems Workshop (MASS'07)*, pages 1–7, 2007.

[18] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac : a short preamble mac protocol for duty-cycled wireless sensor networks. In *Proc. of the 4th international conference on Embedded networked sensor systems (SenSys '06)*, pages 307–320, New York, NY, USA, 2006.

[19] Yang Chen, Omprakash Gnawali, Maria Kazandjieva, Philip Levis, and John Regehr. Surviving sensor network software faults. In *SOSP '09 : Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, pages 235–246, New York, NY, USA, 2009. ACM.

[20] C.T. Chou, J. Yang, and D. Wang. Cooperative mac protocol with automatic relay selection in distributed wireless networks. In *Proc. of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW '07)*, pages 526–531, 2007.

[21] T. Clausen and P. Jacquet. *Optimized Link State Routing Protocol (OLSR) RFC 3626.* 2003.

[22] Code Blue Project, Harvard Sensor Network Lab. http ://fiji.eecs.harvard.edu/CodeBlue.

[23] S. Coleri and P. Varaiya. PEDAMACS : Power efficient and delay aware medium access protocol for sensor networks. *IEEE Transactions on Mobile Computing*, 5(7) :920–930, 2006.

[24] S. Coyle, Y. Wu, K.-T. Lau, S. Brady, G. Wallace, and D. Diamond. Bio-sensing textiles - wearable chemical biosensors for health monitoring. In *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN'07)*, pages 35–39. SpringerLink, March 2007.

[25] D. Cypher, N. Chevrollier, N. Montavont, and N. Golmie. Prevailing over wires in healthcare environments : Benefits and challenges. *Communications Magazine, IEEE*, 44(4) :56–63, April 2006.

[26] F. Dabiri, H. Noshadi, H. Hagopian, T. Massey, and M. Sarrafzadeh. Light-weight medical bodynets. In *Proc. of the Second International Conference on Body Area Network (BodyNets'07)*, 2007.

[27] T. Dierks and E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346.* 2006.

[28] M. Dohler, A. Gkelias, and A.H. Aghvami. Capacity of distributed phy-layer sensor networks. *IEEE Trans. on Vehicular Technology*, 55(2) :622–639, March 2006.

[29] D. Dolev and A. Yao. On the security of public key protocols. In *IEEE Transactions on Information Theory*, volume 29, pages 198–208, 1983.

[30] A. El-Hoiydi and J. D. Decotignie. Wisemac : an ultra low power mac protocol for the downlink of infrastructure wireless sensor networks. volume 1, pages 244–251 Vol.1, 2004.

[31] Eyes, Energy efficient sensor networks. http ://www.eyes.eu.org/index.htm.

[32] Y. Fan and J. Thompson. Mimo configurations for relay channels : Theory and practice. *IEEE Trans. on Wireless Comm.*, 6(5) :1774–1786, May 2007.

[33] S. Fouladgar, B. Mainaud, K. Masmoudi, and H. Afifi. Tiny 3-tls : A trust delegation protocol for wireless sensor networks. In *Proc. of the Third European workshop on Security and Privacy in Ad-Hoc and Sensor Networks (ESAS'06)*, pages 32–42, 2006.

[34] T. Gao, C. Pesto, L. Selavo, Y. Chen, J. G. Ko, J. H. Lim, A. Terzis, A. Watt, J. Jeng, B. R. Chen, K. Lorincz, and M. Welsh. Wireless medical sensor networks in emergency response : Implementation and pilot results. *Proc. of the IEEE Conference on Technologies for Homeland Security (HST'08)*, pages 187–192, May 2008.

[35] M.C. Ghedira, B. Mainaud, H. Afifi, and A. Belghith. Improving bandwidth utilization in medical sensor networks. In *Proc. of Applications and Services in Wireless Networks (ASWN'06)*, pages 35–40, 2006.

[36] N. Golmie, D. Cypher, and O. Rebala. Performance evaluation of low rate WPANs for medical applications. In *Proc. of IEEE Military Communications Conference (MILCOM'04)*, pages 927–933, 2004.

[37] V. Gupta, M. Millard, S. Fung, Y. Zhu, N. Gura, H. Eberle, and S. C. Shantz. Sizzle : A standards-based end-to-end security architecture for the embedded internet. In *Proc. of the Third IEEE International Conference on Pervasive Computing and Communications (PERCOM'05)*, pages 247–256, 2005.

[38] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. In *Proc of of the 6th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'04)*, pages 119–132, 2004.

[39] H. Hassanein and J. Luo. Reliable energy aware routing in wireless sensor networks. In *Proc. of the Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems (DSSNS'06)*, pages 54–64, 2006.

[40] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. of the 33rd Hawaii International Conference on System Sciences (HICSS'00)-Volume 8*, 2000.

[41] R.N. Horspool and W.J. Windels. An lz approach to ecg compression. *Computer-Based Medical Systems, 1994., Proceedings 1994 IEEE Seventh Symposium on*, pages 71–76, Jun 1994.

[42] Chakeres I and Perkins C. Dynamic manet on-demand (dymo) routing, draft-ietf-manet-dymo-12, expires : August 10, 2008.

[43] IEEE Standards Board. IEEE standard glossary of software engineering terminology— IEEE std 610.12-1990 (r2002), 2002.

[44] IEEE std 802.15.4-2009 (revision of ieee std 802.15.4-2003). 2006.

[45] Intel Mote,. http ://www.intel.com/research/exploratory/motes.htm.

[46] Intel's Proactive Health. . http ://www.intel.com/healthcare/hri/pdf/proactive_health.pdf.

[47] W. Ji and B. Y. Zheng. A novel cooperative MAC protocol for WSN based on NDMA. In *Proc. of The 8th International Conference on Signal Processing (ICSP'06)*, pages 16–20, 2006.

[48] H. Ju and L. Cui. EasiPC : a packet compression mechanism for embedded WSN. In *Proc. of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05)*, pages 394–399, 2005.

[49] A. Lacombe, F. Rocaries, C. Dietrich, Jean-Louis Baldinger, Jérôme Boudy, François Delavault, A. d'Escatha, M. Baer, and A. Ozguler. Open Technical Prototype and Validation Process model for Patient at Home Medical Monitoring system. In *Proc. of Modeling and Simulation in Biology, Medicine and Biomedical Engineering (Biomedsim'05)*, 2005.

[50] J. N. Laneman, D. N. C. Tse, and G. W. Wornell. Cooperative diversity in wireless networks : Efficient protocols and outage behavior. *IEEE Trans. on Inf. Theory*, 50(12) :3062–3080, Dec. 2004.

[51] J. N. Laneman and G. W. Wornell. Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks. *IEEE Trans. on Inf. Theory*, 49(10) :2415–2425, 2003.

[52] K. Langendoen. Medium access control in wireless networks. In *Medium Access Control in Wireless Networks, Volume II : Practice and Standards*. Nova Science Publishers, 2008.

[53] Y.-Z. Lee, M. Gerla, J. Chen, J. Chen, B. Zhou, and A. Caruso. Direction forward routing for highly mobile ad hoc networks. *Ad Hoc and Sensor Wireless Networks*, 2(2), 2006.

[54] Philip Levis. Low-Power Sensor Networks : A Case Study in Seeking Distributed Dependability ; Keynote at NSF HCCS-CPS Workshop, 2006.

[55] Dan Li and David R. Cheriton. Evaluating the utility of fec with reliable multicast. In *ICNP '99 : Proceedings of the Seventh Annual International Conference on Network Protocols*, page 97, Washington, DC, USA, 1999. IEEE Computer Society.

[56] Yun Li, Hao Jia, Zhan jun Liu, Zhi Ren, and Guo jun Li. A mac protocol based on s-mac for power asymmetric wsn network. In *Computer Science and Computational Technology, 2008. ISCSCT '08. International Symposium on*, volume 1, pages 299–302, Dec. 2008.

[57] R. Lin and A.P. Petropulu. A new wireless network medium access protocol based on cooperation. *IEEE Trans. on Signal Processing*, 53(12) :4675–4684, Dec. 2005.

[58] J. D. C. Little. A Proof of the Queueing Formula $L = \lambda W$. *Operations Research*, 9 :380–387, 1961.

[59] P. Liu, Z. Tao, and S. Panwar. A cooperative MAC protocol for wireless local area networks. In *Proc. of the International Conference on Communications (ICC 2005)*, volume 5, pages 2962–2968, Seoul, Korea, 2005.

[60] B. Lo and G. Z. Yang. Key technical challenges and current implementations of body sensor networks. In *Proc. of the 2nd International Workshop on Body Sensor Networks (BSN'05)*, 2005.

[61] B. P. L. Lo, S. Thiemjarus, R. King, and G. Z. Yang. Body sensor network - a wireless sensor platform for pervasive healthcare monitoring. In *Proc. of the 3rd International conference on Pervasive Computing (PERVASIVE'05*, pages 77–80, 2005.

[62] A. G. Longley and P. L. Rice. Prediction of tropospheric transmission loss over irregular terrain, a computer method. Technical Report ESSA 79-ITSS 67, U.S. Dept. of Commerce, Office of Telecommunications, Boulder, CO, July 1968.

[63] B. Mainaud, V. Gauthier, and H. Afifi. Cooperative communication for wireless sensors network : A mac protocol solution. In *Wireless Days, 2008. WD '08. 1st IFIP*, pages 1–5, Nov. 2008.

[64] Bastien Mainaud, Mariem Zekri, and Hossam Afifi. Improving routing reliability on wireless sensors network with emergency paths. In *ICDCSW '08 : Proceedings of the 2008 The 28th International Conference on Distributed Computing Systems Workshops*, pages 545–550, Washington, DC, USA, 2008. IEEE Computer Society.

[65] V. S. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology - (CRYPTO '85)*, pages 417–426, 1985.

[66] S. Moh, C. Yu, S.-M. Park, H.-N. Kim, and J. Park. Cd-mac : Cooperative diversity mac for robust communication in wireless ad hoc networks. In *Proc. of the International Conference on Communications (ICC '07)*, pages 3636–3641, Glasgow, Scotland, June 2007.

[67] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. *Transmission of IPv6 Packets over IEEE 802.15.4 Networks RFC 4944.* 2007.

[68] Sentilla - Ex-Moteiv, Pervasive Computing Solutions. http ://www.sentilla.com.

[69] R.N. Murty, G. Mainland, I. Rose, A.R. Chowdhury, A. Gosain, J. Bers, and M. Welsh. Citysense : An urban-scale wireless sensor network and testbed. pages 583–588, May 2008.

[70] Personal Navigation and Information System for Users of Public Transport. http ://virtual.vtt.fi/virtual/noppa/noppaeng.htm.

[71] R. Ogier, F. Templin, and M. Lewis. *RFC 3684, Topology Dissemination Based on Reverse-Path Forwarding (TBRPF).* RFC Editor, United States, 2004.

[72] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov. System architecture of a wireless body area sensor network for ubiquitous health monitoring. In *Journal of Mobile Multimedia 1(4)*, pages 307–326, 2006.

[73] S.-J. Park and R. Sivakumar. Sink-to-sensors reliability in sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3) :27–28, 2003.

[74] Seung-Jong Park, R. Sivakumar, I.F. Akyildiz, and R. Vedantham. Garuda : Achieving effective reliability for downstream communication in wireless sensor networks. *Mobile Computing, IEEE Transactions on*, 7(2) :214–230, Feb. 2008.

[75] C. Perkins. Ad-hoc on-demand distance vector routing, in MILCOM '97 panel on Ad Hoc Networks, Nov. 1997.

[76] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *SIGCOMM '94 : Proceedings of the conference on Communications architectures, protocols and applications*, volume 24, pages 234–244. ACM Press, October 1994.

[77] Marina Petrova, Lili Wu, Petri Mahonen, and Janne Riihijarvi. Interference measurements on performance degradation between colocated ieee 802.11g/n and ieee 802.15.4 networks. In *Proc. of the Sixth International Conference on Networking (ICN'07)*, page 93, Washington, DC, USA, 2007. IEEE Computer Society.

[78] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proc. of the 2nd international conference on Embedded networked sensor systems (SenSys '04)*, pages 95–107, New York, NY, USA, 2004.

[79] MATLAB R2007a. http ://www.mathworks.com, 2007.

[80] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *ACM journal on Wireless Networks*, 12(1) :63–78, February 2006.

[81] B. Raman, K. Chebrolu, N. Madabhushi, D. Y. Gokhale, P. K. Valiveti, and D Jain. Implications of link range and (in)stability on sensor network architecture. In *Proc. of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization (WinTECH '06)*, pages 65–72, New York, NY, USA, 2006. ACM.

[82] C. De Santis, B. Mainaud, and H. Afifi. Expérimentations et programmation des réseaux de capteurs ieee 802.15.4. Technical Report 06010-RS2M, Institut national des télécommunications, Evry, France, December 2006.

[83] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizing sensor networks in the energy-latency-density design space, 2002.

[84] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, and J. Porter. Luster : wireless sensor network for environmental research. In *SenSys '07 : Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 103–116, New York, NY, USA, 2007. ACM.

[85] E. Shih, V. Bychkovsky, D. Curtis, and J. Guttag. Continuous medical monitoring using wireless microsensors. In *Proc. of the 2nd international conference on Embedded networked sensor systems (Sensys'04)*, pages 310–310, 2004.

[86] Victor Shnayder, Bor-rong Chen, Konrad Lorincz, Thaddeus R. F. Fulford Jones, and Matt Welsh. Sensor networks for medical care. In *Proc. of the 3rd international conference on Embedded networked sensor systems (SenSys'05)*, pages 314–314, New York, NY, USA, 2005. ACM.

[87] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. In *Proc of the 4th international conference on Embedded networked sensor systems (SenSys '06)*, pages 419–420, New York, NY, USA, 2006. ACM.

[88] K. Srinivasan and P. Levis. Rssi is under appreciated. In *Proc. of the Third Workshop on Embedded Networked Sensors (EmNets'06)*, 2006.

[89] F. Stann and J. Heidemann. Rmst : reliable data transport in sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 102–112, May 2003.

[90] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An analysis of a large scale habitat monitoring application. In *Proc. of the 2nd international conference on Embedded networked sensor systems (SenSys'04)*, pages 214–226, 2004.

[91] Igor Talzi, Andreas Hasler, Stephan Gruber, and Christian Tschudin. Permasense : investigating permafrost with a wsn in the swiss alps. In *EmNets '07 : Proceedings of the 4th workshop on Embedded networked sensors*, pages 8–12, New York, NY, USA, 2007. ACM.

[92] N.F. Timmons and W.G. Scanlon. Analysis of the performance of ieee 802.15.4 for medical sensor body area networking. *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 16–24, Oct. 2004.

[93] TinyOS, An open-source Operating System for Wireless Embedded Sensor Networks. http ://www.tinyos.net.

[94] L. Tong, Q. Zhao, and G. Mergen. Multipacket reception in random access wireless networks : From signal processing to optimal medium access control. *IEEE Communication magazine*, 39(11) :108–112, Nov 2001.

[95] Tymo, An implementation of the DYMO protocol on TinyOS. http ://tymo.sourceforge.net/news/.

[96] T. van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proc of the 1st international conference on Embedded networked sensor systems (SenSys '03)*, pages 171–180, New York, NY, USA, 2003. ACM Press.

[97] L. van Hoesel and P. Havinga. A lightweight medium access protocol (LMAC) for wireless sensor networks : reducing preamble transmissions and transceiver state switches. In *Proc. of the International Conference on Networked Sensing Systems (INSS'04)*, Tokyo, Japan, 2004.

[98] S. Varshney, U. ; Sneha. Patient monitoring using ad hoc wireless networks : reliability and power management. *Communications Magazine, IEEE*, 44(4) :49–55, April 2006.

[99] J-P. Vasseur. Terminology in low power and lossy networks, draft-ietf-roll-terminology-01.txt, expires : November 8, 2009.

[100] L. Viganò. Automated security protocol analysis with the avispa tool. In *Proc. of the 21st Annual Conference on Mathematical Foundations of Programming Semantics (MFPS'06)*, pages 61–86, 2006.

[101] A.B. Waluyo, I. Pek, Song Ying, Jiankang Wu, Xiang Chen, and Wee-Soon Yeoh. Litemwban : A lightweight middleware for wireless body area network. pages 141–144, June 2008.

[102] Chieh-Yih Wan, Andrew T. Campbell, and Lakshman Krishnamurthy. Psfq : a reliable transport protocol for wireless sensor networks. In *WSNA '02 : Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 1–11, New York, NY, USA, 2002. ACM.

[103] Ronald Watro, Derrick Kong, Sue-fen Cuti, Charles Gardiner, Charles Lynn, and Peter Kruus. Tinypk : securing sensor networks with public key technology. In *SASN '04 : Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 59–64, New York, NY, USA, 2004. ACM.

[104] Webs, Wireless Embedded Systems Department. University of California, Berkeley. http ://local.cs.berkeley.edu/webs.

[105] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proc. of the 1st international conference on Embedded networked sensor systems (SenSys '03)*, pages 14–27, New York, NY, USA, 2003. ACM Press.

[106] Ultra Low Power Module for sensor network applications,. http ://www.intel.com/research/exploratory/motes.htm.

[107] CrossBow, Wireless sensor networks. http ://www.xbow.com.

[108] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks, 2002.

[109] W. Ye, F. Silva, and J. Heidemann. Ultra-low duty cycle mac with scheduled channel polling. In *Proc. of the 4th international conference on Embedded networked sensor systems (SenSys '06)*, pages 321–334, New York, NY, USA, 2006. ACM Press.

[110] Dae Gil Yoon, Soo Young Shin, Wook Hyun Kwon, and Hong Seong Park. Packet error rate analysis of ieee 802.11b under ieee 802.15.4 interference. *IEEE Vehicular Technology Conference. VTC 2006-Spring*, 3 :1186–1190, May 2006.

[111] M. R. Yuce, P. C. Ng, and J. Y. Khan. Monitoring of physiological parameters from multiple patients using wireless sensor network. *Journal of Medical Systems*, 32(5) :433–441, 2008.

[112] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3) :337–343, 1977.