

Table des matières

| | |
|---|-----------|
| Notations | 1 |
| Introduction Générale | 3 |
| I CONTEXTE ET MOTIVATIONS | 7 |
| I.1 Introduction sur les réseaux de capteurs sans fil | 8 |
| I.1.1 Généralités | 8 |
| I.1.2 Réseaux de capteurs sans fil | 8 |
| I.1.2.1 Mise en œuvre | 10 |
| I.1.2.2 Déploiement | 11 |
| I.1.2.3 Exemple d'applications des WSNs | 11 |
| I.1.3 Le nœud capteur | 12 |
| I.1.3.1 Composants matériels et logiciels | 12 |
| I.1.3.2 Modules de nœuds capteurs existants | 14 |
| I.1.3.3 La pile protocolaire | 16 |
| I.2 Principales caractéristiques d'un WSN | 18 |
| I.2.1 Les réseaux traditionnels vs les WSNs | 19 |
| I.2.1.1 Typologie des réseaux sans fil | 19 |
| I.2.1.2 Les principales normes de communication sans fil | 20 |
| I.2.1.3 Protocoles de communication pour les WSNs | 23 |
| I.2.2 La limitation en énergie pour un WSN | 25 |
| I.2.3 La mobilité | 27 |
| I.3 Optimisation et paramètres de performances | 28 |
| I.3.1 L'optimisation à travers les principes du cross-layer | 28 |
| I.3.2 Les principaux paramètres de performances | 29 |
| I.3.2.1 La gestion du réseau | 30 |
| I.3.2.2 La synchronisation | 31 |
| I.3.2.3 La synchronisation vs l'économie d'énergie | 33 |
| I.3.2.4 L'économie d'énergie | 34 |
| I.3.2.5 Topologie dynamique et passage à l'échelle | 35 |
| I.4 Evaluation des WSNs | 36 |
| I.4.1 Etude des outils disponibles | 37 |
| I.4.2 Justification de notre choix | 42 |
| I.5 Objectifs de notre recherche | 43 |
| I.5.1 Présentation générale de notre application | 44 |
| I.5.2 Le choix de la topologie | 45 |
| I.5.3 Le besoin de synchronisation | 46 |
| I.5.4 Le besoin d'économie d'énergie | 47 |
| I.6 Conclusion | 50 |
| II CONCEPTION ET MODELISATION D'UN RESEAU D'INSTRUMENTATION ET DE MESURES SANS FIL | 53 |
| II.1 Introduction | 53 |
| II.2 Description de notre réseau d'instrumentation et de mesures | 55 |
| II.2.1 Topologie du réseau | 55 |
| II.2.2 Méthodologie et choix de technologies | 56 |
| II.2.3 Spécification des besoins | 58 |
| II.2.4 Analyse préliminaire du système | 58 |
| II.2.5 Conception du système | 63 |
| II.2.5.1 Description des couches liaison de données et physique | 64 |
| II.2.5.2 Description de la couche réseau | 67 |

| | | |
|--------------|---|------------|
| II.2.5.3 | Description de la couche Applicative | 69 |
| II.3 | Description de la plateforme matérielle | 72 |
| II.3.1 | Carte Wisair Wireless USB PCI Express Mini | 72 |
| II.3.2 | Carte NI PCI-1588..... | 73 |
| II.3.3 | Plateforme ML410..... | 74 |
| II.4 | Modélisation sur NS-2 | 75 |
| II.4.1 | Eléments de base de la simulation en NS-2 | 76 |
| II.4.2 | Les principales contributions en NS-2 | 78 |
| II.4.2.1 | Implémentation de notre topologie réseau..... | 79 |
| II.4.2.2 | Proposition d'utilisation d'interfaces multiples..... | 80 |
| II.5 | Conclusion | 81 |
| III | PROPOSITION D'UN NOUVEAU PROTOCOLE DE SYNCHRONISATION | 83 |
| III.1 | Introduction..... | 83 |
| III.2 | Problématique de la synchronisation | 84 |
| III.2.1 | Stratégies possibles..... | 84 |
| III.2.2 | Analyse..... | 85 |
| III.2.3 | Bilan | 85 |
| III.2.4 | Le standard IEEE-1588 | 86 |
| III.2.5 | Le protocole de synchronisation PBS..... | 88 |
| III.3 | Présentation de la contribution: l'IEEE1588-PBS hybride | 90 |
| III.3.1 | Algorithme et fonctionnement | 91 |
| III.3.2 | Intégration dans notre cas d'étude..... | 92 |
| III.3.3 | L'impact sur la consommation d'énergie..... | 94 |
| III.4 | Implémentation et mise en œuvre | 94 |
| III.4.1 | Conditions pratiques de mise en œuvre | 94 |
| III.4.2 | Les étapes d'implémentation dans le simulateur NS-2 | 95 |
| III.4.2.1 | L'intégration du modèle d'horloge interne | 95 |
| III.4.2.2 | La phase d'implémentation d'IEEE-1588..... | 96 |
| III.4.2.3 | La phase d'implémentation d'IEEE1588-PBS hybride..... | 98 |
| III.5 | Conclusion | 100 |
| IV | LES ANTENNES DIRECTIVES DANS LES RESEAUX DES CAPTEURS SANS FIL .. | 101 |
| IV.1 | Introduction..... | 101 |
| IV.2 | L'impact des antennes directives sur les différentes couches de communication..... | 103 |
| IV.2.1 | Introduction..... | 103 |
| IV.2.2 | Concepts de base | 104 |
| IV.2.3 | Propositions pour l'utilisation d'antennes directives..... | 107 |
| IV.2.3.1 | Solutions existantes pour la couche MAC | 109 |
| IV.2.3.2 | Les principaux protocoles de routage | 111 |
| IV.3 | L'applicabilité dans une catégorie large de réseaux sans fil .. | 113 |
| IV.4 | Principe et implémentation de notre solution | 114 |
| IV.4.1 | Proposition d'architecture d'antennes directives | 115 |
| IV.4.2 | L'implémentation de notre solution | 117 |
| IV.5 | Conclusion | 118 |
| V | SIMULATIONS ET EXPERIMENTATIONS | 119 |
| V.1 | Introduction..... | 119 |
| V.2 | Topologie : choix et validation..... | 120 |
| V.2.1 | Objectif : déterminer la meilleure topologie du système..... | 120 |
| V.2.2 | Intégration des interfaces multiples | 122 |
| V.2.3 | Synthèse..... | 127 |
| V.3 | La synchronisation vs. l'économie d'énergie..... | 127 |
| V.3.1 | Evaluation des solutions existantes | 128 |
| V.3.1.1 | Description de la plateforme | 128 |
| V.3.1.2 | Etudes sur les performances de synchronisation | 130 |
| V.3.1.3 | Conclusion..... | 137 |
| V.3.2 | Notre proposition de synchronisation. Analyse comparative des performances..... | 138 |

| | | |
|--|---|------------|
| V.3.2.1 | La précision de la synchronisation..... | 139 |
| V.3.2.2 | Evaluation de la consommation d'énergie..... | 141 |
| V.3.3 | Conclusion..... | 142 |
| V.4 | La directivité des antennes vs. l'économie d'énergie | 142 |
| V.4.1 | Analyse comparative des performances..... | 142 |
| V.4.2 | Conclusion..... | 146 |
| V.5 | Synthèse générale..... | 146 |
| Conclusion Générale et Perspectives | | 147 |
| Bibliographie | | 151 |
| Table des figures | | 161 |
| Liste des tableaux..... | | 163 |

Notations

| | |
|----------|--|
| ABR | Associativity Based Routing |
| AODV | Ad hoc On-demand Distance Vector Routing |
| API | Application Programming Interface |
| ASIC | Application-Specific Integrated Circuits |
| BER | Bet Error Rate |
| CAN | Controller Area Network |
| CBQ | Class-Based Queueing |
| CSMA/CA | Carrier Sense Multiple Access with Collision Avoidance |
| CSMA/CD | Carrier Sense Multiple Access with Collision Detection |
| CSP | Clock Synchronization Precision |
| DCF | Distributed Coordination Function |
| DIFS | Distributed Inter Frame Spacing |
| DNAV | Directional Network Allocation Vector |
| DropTail | Simple queue management algorithm used by internet routers |
| DRR | Dynamic Rate Repartitioning |
| DSDV | Destination Sequenced Distance-Vector |
| DSP | Digital Signal Processing |
| DSR | Dynamic Source Routing Protocol |
| DVCS | Directional Virtual Carrier Sensing |
| DVS | Dynamic Voltage Scaling |
| EEPROM | Electrically Erasable Programmable Read-Only Memory |
| eTPU | Enhanced Time Processing Unit |
| FEC | Forward Error Correction |
| FPGA | Field-Programmable Gate Arrays |
| FSR | Fisheye State Routing |
| Ftp | File Transfer Protocol |
| FTSP | Flooding Time Synchronization Protocol |
| LAN | Large Area Network |
| LR-WPAN | Low Rate Wireless Personal Area Network |
| LUT | Look-Up Table |
| MAC | Media Access Control |
| MANET | Mobile Ad Hoc Network |
| NAM | Network Animator |
| QoS | Quality of Service |
| OFDM | Orthogonal Frequency Division Multiplexing |
| OLSR | Optmised Link State Routing |
| OTcl | Object-oriented Tool Command Language |
| OWD | One-Way-Delay |
| PBS | Pairwise Broadcast Synchronization |

| | |
|---------|---|
| PCS | Physical Carrier Sensing |
| PDU | Protocol Data Unit |
| PFSQ | Pump slowly, Fetch quickly |
| PTP | Precision Time Protocol |
| RAM | Random Access Memory |
| RBS | Reference Broadcast Synchronization |
| RED | Random Early Detection |
| RF | Radio Frequency |
| ROM | Read-Only Memory |
| ROS | Receiver-Only Synchronization |
| RTP | Real-time Transport Protocol |
| RTS/CTS | Request to Send / Clear to Send |
| SAR | Sequential Assignment Routing |
| SIFS | Short Inter Frame Spacing |
| S-MAC | Sensor – MAC |
| SMACS | Self-Organizing Medium Access Control for Sensors |
| SMECN | Small Minimum Energy Communication Network |
| SNR | Signal-to-Noise Ratio |
| SPIN | Sensor Protocol for Information via Negotiation |
| SRM | Storage/ Retrieval Machines |
| SSR | Signal Stability Routing |
| Tcl | Terminal Control Language |
| TclCL | Tcl with Classes |
| TCP | Transmission Control Protocol |
| TCXO | Temperature Compensated Crystal Oscillator |
| Telnet | TERminal NETwork |
| TDMA | Time Division Multiple Access |
| TORA | Temporally Ordered Routing Algorithm |
| TPSN | Timing-sync Protocol for Sensor Networks |
| TSBS | Time-Stamp Broadcast Synchronization |
| UCT | Universal Coordinated Time |
| UDP | User Datagram Protocol |
| UWB | Ultra Wide Band |
| WEB | World Wide Web |
| WI-FI | Wireless Fidelity |
| WRP | Wireless Routing Protocol |
| WSN | Wireless Sensor Network |

Introduction Générale

Dans les années 1990, dans le monde de la recherche, est apparue une idée qui paraissait plutôt un rêve pour cette époque : imaginer « un système nerveux central pour la Terre », capable de surveiller en temps réel les événements, ayant comme principaux bénéfices de pouvoir empêcher les accidents et d'économiser l'énergie. « Cette poussière intelligente a mis longtemps à apparaître » dit le professeur Pister, de l'Université de Californie à Berkeley. « J'ai inventé l'expression il y a 14 ans. La poussière vraiment futée a mis le temps, mais elle est finalement arrivée ».

Aujourd'hui les réseaux de capteurs sont devenus des systèmes pouvant atteindre un très grand nombre de nœuds, avec une zone de couverture déterminée et déployés d'une manière plus ou moins dense dans un environnement hétérogène dont on mesure ainsi son état global. Les derniers progrès en terme de miniaturisation, ainsi que le remplacement du câblage classique par des technologies de communication radio, ont généré de nouvelles catégories d'applications qui visent de nombreux domaines : l'aéronautique, l'automobile, le médical, l'environnement, etc. De plus, les progrès des communications sans fil permettent aujourd'hui de répondre à des exigences peu envisageables auparavant.

Le côté spécifique de ces réseaux est notamment leur capacité d'auto-organisation et d'auto-adaptation. Cette caractéristique est utilisée pour le déploiement aléatoire des nœuds capteurs, dictée par le type d'application (par exemple la surveillance/détection d'incendie dans une forêt). Dans de telles conditions d'accès difficile où il est peu envisageable d'avoir accès à une source d'énergie, la durée de vie d'un nœud capteur va se limiter à la durée de vie de sa batterie. Pour cela, une des principales préoccupations dans un réseau de capteurs sans fil est la gestion d'énergie qui implique des efforts d'optimisation depuis la couche physique jusqu'au niveau applicatif.

LA PROBLEMATIQUE ADRESSEE

Une fois placées dans ce contexte général, nous pouvons énoncer la problématique de cette thèse, qui consiste à concevoir une architecture pour les objets communicants à faible consommation en utilisant des antennes « intelligentes » pour l'instrumentation et la mesure. Intégrant une approche pluridisciplinaire, cette architecture couvre les services offerts depuis les couches MAC jusqu'à celles de plus haut niveau. Basés sur une partie matérielle complètement reconfigurable (amplificateur de puissance et antennes à base de MEMS RF), les services des couches supérieures sont définis en partie sur circuits numériques pour la couche physique (bande de base) et la couche MAC, et de manière logicielle pour les protocoles de routages adaptés et les services innovants.

Il s'agit de ce fait de concevoir un réseau de centaines de capteurs sans fil qui peut avoir diverses applications, parfois même embarquées comme en aéronautique, spatial, automobile ou ferroviaire. Ainsi, l'application qui nous sert de référence principale a pour but de mettre en œuvre des capteurs de température et de pression sur les ailes d'un avion.

Une architecture de communication hiérarchique telle que celle proposée dans notre application, s'appuie sur un réseau de communication à deux niveaux :

- ❖ Un premier niveau passif, capable d'interroger des capteurs autonomes à l'aide de technologies de communication peu consommatrices d'énergie (radars, RFID, etc.) ;
- ❖ Un second niveau qui intègre les fonctionnalités d'interconnexion et de coordination nécessaires à l'intégration de plusieurs réseaux de premier niveau.

En effet, il semble difficile de multiplier indéfiniment le nombre de capteurs passifs avec les moyens d'adressage limités de cette technologie (multiplexage fréquentiel). Le rôle d'une couche de coordination intégrant des fonctionnalités d'adressage et de routage est ainsi indispensable pour permettre le passage à l'échelle et intégrer au sein d'un même réseau plusieurs milliers de nœuds.

En résumé, notre travail consiste à concevoir un système autonome multi capteurs, d'acquisition et de traitement avec mémorisation, communiquant à travers un réseau sans fil. Parmi les principaux problèmes à résoudre, nous trouvons :

- ❖ Le contrôle de la topologie ;
- ❖ La précision de la synchronisation ;
- ❖ La consommation d'énergie.

LES PRINCIPALES CONTRIBUTIONS

Afin de répondre aux problèmes annoncés ci-dessus, en nous basant sur la conception de la couche PHY et de la couche MAC, avec les organes de transmission tels que les antennes émettrices, nous apportons une première amélioration par la proposition d'une architecture hiérarchique pour le contrôle de la topologie. Une autre amélioration envisagée concerne l'utilisation des interfaces multiples pour les éléments du réseau. Nous évaluerons la performance de cette solution par simulation à l'aide de l'outil que nous avons choisi : le simulateur NS-2. On pourra ainsi montrer une réduction de la consommation d'énergie en utilisant les interfaces multiples.

Une deuxième proposition consiste à améliorer la précision demandée pour la synchronisation des données, qui, dans l'application visée, est une contrainte importante. Ce travail réside d'abord en une étape de test du matériel existant, i.e. des cartes NI PCI-15881 basées sur la norme IEEE-1588, et par la suite en la conception d'une solution de synchronisation mieux adaptée. Les résultats de simulations montrent que notre proposition est efficace non seulement en termes de précision de la synchronisation, mais aussi en ce qui concerne la consommation d'énergie, en comparaison à d'autres approches trouvées dans la littérature.

Enfin, nous avons proposé d'améliorer la gestion de la consommation d'énergie de notre architecture en utilisant des antennes directionnelles. Pour cela, nous avons développé un modèle qui nous permet d'évaluer les performances de la solution proposée. Les simulations que nous avons réalisées ont ainsi montré que, en plus des avantages apportés par les antennes directives en termes de couverture, collisions, etc., une réduction importante de la consommation d'énergie est enregistrée.

De ce fait, le manuscrit de thèse est organisé de la manière suivante :

- ❖ Le premier chapitre est consacré à une présentation générale des réseaux de capteurs sans fil, avec leurs modèles et composants caractéristiques, ainsi que les applications associées. Nous rappelons également quelles sont les principales problématiques associées, dans le but de pouvoir proposer les solutions

¹ http://www.ni.com/pdf/products/us/cat_pci1588.pdf

d'optimisation les mieux adaptées. En vue de l'évaluation des réseaux de capteurs sans fil, nous réalisons une étude des simulateurs existants, afin de tester leur fonctionnement dans des conditions réelles.

- ❖ Le chapitre suivant présente les spécificités d'une application d'instrumentation et de mesures et parcourt les étapes de conception qui sont nécessaires pour la mise en œuvre et l'optimisation du système. Nous abordons notamment le choix des technologies et des protocoles les plus adaptés dans ce contexte pour répondre aux besoins et proposer des solutions d'optimisation.
- ❖ Le chapitre III décrit un nouveau protocole qui a pour but de garantir une bonne précision de synchronisation, tout en visant la minimisation de la consommation d'énergie.
- ❖ Le chapitre IV traite enfin plus précisément des antennes directives ou intelligentes dans les réseaux d'instrumentation et de mesures en démontrant les avantages de cette nouvelle proposition, notamment sur l'aspect consommation d'énergie et sur la qualité de service.
- ❖ Le dernier chapitre décrit et analyse les résultats obtenus lors de nos simulations et expérimentations avant de conclure sur les perspectives ouvertes par les travaux réalisés dans cette thèse.

I CONTEXTE ET MOTIVATIONS

| | | |
|------------|--|-----------|
| I.1 | Introduction sur les réseaux de capteurs sans fil | 8 |
| I.1.1 | Généralités | 8 |
| I.1.2 | Réseaux de capteurs sans fil | 8 |
| I.1.3 | Le nœud capteur | 12 |
| I.2 | Principales caractéristiques d'un WSN | 18 |
| I.2.1 | Les réseaux traditionnels vs les WSNs | 19 |
| I.2.2 | La limitation en énergie pour un WSN | 25 |
| I.2.3 | La mobilité | 27 |
| I.3 | Optimisation et paramètres de performances | 28 |
| I.3.1 | L'optimisation à travers les principes du cross-layer | 28 |
| I.3.2 | Les principaux paramètres de performances | 29 |
| I.4 | Evaluation des WSNs | 36 |
| I.4.1 | Etude des outils disponibles | 37 |
| I.4.2 | Justification de notre choix | 42 |
| I.5 | Objectifs de notre recherche | 43 |
| I.5.1 | Présentation générale de notre application | 44 |
| I.5.2 | Le choix de la topologie | 45 |
| I.5.3 | Le besoin de synchronisation | 46 |
| I.5.4 | Le besoin d'économie d'énergie | 47 |
| I.6 | Conclusion | 50 |

Les réseaux de capteurs sans fil sont des systèmes composés d'un grand nombre de nœuds individuels, qui sont capables d'interagir avec leur environnement par la détection ou le contrôle de paramètres physiques. La collaboration entre tous ces nœuds est assurée par une communication sans fil, qui apporte des multiples avantages (mobilité, déploiement et maintenance faciles, etc.). En contrepartie, les défis sont aussi nombreux (optimiser l'autonomie des nœuds, assurer la couverture et une connectique fiable, optimiser la précision des données transmises, etc.).

Dans ce premier chapitre, nous allons faire une présentation générale des réseaux de capteurs sans fil, avec leurs modèles et composants caractéristiques, ainsi que les applications associées. Cette partie donne aussi un aperçu de haut niveau sur la conception de l'architecture de communication de ces réseaux, directement liée au type d'application. Ainsi, l'objectif de ce chapitre sera d'étudier et d'analyser cette catégorie de systèmes avec leurs paramètres associés, dans le but de réaliser une évaluation significative de ce type de réseau.

I.1 Introduction sur les réseaux de capteurs sans fil

I.1.1 Généralités

Par l'intégration des notions de calcul et de contrôle dans notre environnement physique, les paradigmes d'interaction comme « person to person », « person to machine » et « machine to machine » peuvent être complétés par celui de « person to physical ». Ainsi, ce dernier devient plus important qu'une simple manipulation symbolique de données [1].

La technologie des capteurs existant aujourd'hui facilite les mesures de paramètres physiques comme la température, l'humidité, la lumière visible et infrarouge, l'acoustique, les vibrations, la pression; il y a aussi des capteurs chimiques, de contraintes mécaniques, magnétiques et même radar [2], [3]. Cette diversité permet le développement des nombreux types d'applications, mais la conception de ces systèmes implique pour la majorité d'entre elles des exigences très différentes et complexes.

Dans un grand nombre d'applications, l'hypothèse d'avoir une interconnexion filaire pour l'ensemble des entités du réseau constitue un obstacle considérable :

- ❖ Étant donné le grand nombre de dispositifs, la complexité du câblage devient trop chère et soulève un problème de maintenance;
- ❖ Cela peut empêcher les entités d'être mobiles, et donc aux actionneurs d'être proches du phénomène qu'ils sont censés contrôler, etc.

En conséquence, la communication sans fil est devenue une nécessité qui a fait l'objet d'une nouvelle classe de réseaux, les réseaux de capteurs sans fil ou les WSNs (Wireless Sensor Networks) [4].

Historiquement, l'apparition de la notion de « nœud capteur » remonte aux années 1990 et est due au chercheur Kris Pister qui avait imaginé « des particules intelligentes » capables de tout surveiller sur la planète. Ainsi, quelques années plus tard, en 1998, c'est le projet SmartDust qui est né, aillant comme objectifs de créer des automates de communication et de détection dans un millimètre cube. Même si le projet a pris fin dès son début, il a donné naissance à d'autres grands projets, qui concernent par exemple les centres de recherche de Berkeley NEST et CENS [5].

Pour concevoir et optimiser un réseau de capteurs sans fil, il faut s'appuyer sur la connaissance et la maîtrise de nombreux domaines : le support de transmission radio avec ses propriétés, la détermination du type de protocole MAC (Media Access Control) le mieux adapté, le choix du protocole de routage en fonction du type d'application, l'alimentation par batterie qui influence les caractéristiques d'un nœud, etc. Cette approche est aussi la stratégie de notre travail, ce qui va nous permettre de concevoir des systèmes plus performants et capables de répondre aux exigences imposées par l'application.

Ainsi, nous présentons par la suite les réseaux de capteurs sans fil d'un point de vue plus général, en s'appuyant sur leurs principales caractéristiques. Ces éléments seront utilisés au long de ce mémoire pour motiver des besoins techniques plus particuliers.

I.1.2 Réseaux de capteurs sans fil

Les réseaux de capteurs sans fil sont des systèmes de communication constitués d'un nombre élevé d'éléments (centaines, voire des milliers d'éléments) appelés nœuds capteurs,

ou par abus de langage, tout simplement nœuds. Un nœud capteur peut être mobile mais aussi bien fixe, tout étant dépendant des besoins de l'application.

Un exemple d'un tel déploiement d'architecture réseau est visible sur la Figure I.1 :

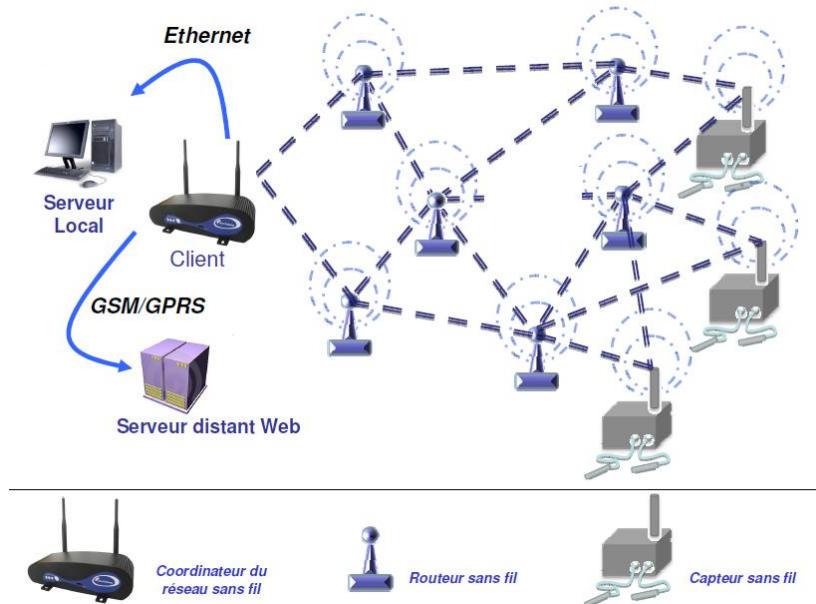


Figure I.1 Exemple d'architecture de réseau de capteurs embarqués sans-fil².

Comme nous pouvons constater, le réseau est suffisamment dense pour permettre une communication multi-saut, ce qui permet aux informations prises par un « capteur sans fil » d'être acheminées jusqu'à la station de base (ici le « coordinateur du réseau sans fil »). Compte tenu du fait que le nœud coordinateur possède des ressources matérielles et énergétiques plus importantes que les autres nœuds, il a le rôle de stocker toutes les informations collectées par les nœuds capteurs.

Les principales caractéristiques d'un WSN sont :

- ❖ La densité élevée due à un grand nombre de capteurs;
- ❖ Le mode de communication basé sur un paradigme centré sur les données car les nœuds capteurs ne sont pas adressés par un identifiant IP mais par un attribut basé, par exemple, sur la localisation ;
- ❖ La limitation en ressources énergétiques ;
- ❖ Le besoin d'une grande précision pour la synchronisation des données ;
- ❖ La capacité d'auto organisation.

En ce qui concerne les applications industrielles, la majorité ont pour but la maîtrise et le contrôle des phénomènes physiques (comme la température, l'altitude, la proximité, etc.) pour le contrôle et la mesure, ce qui a conduit à la naissance d'une catégorie spéciale de réseaux de capteurs sans fil, *les réseaux de capteurs sans fil embarqués*.

Un réseau de capteurs sans fil embarqué est caractérisé par une autonomie d'un point de vue acquisition des informations physiques, traitement, stockage et ensuite transmission de données par un module sans fil.

Etant donné notre cas d'étude (que nous présenterons à la fin de ce chapitre) qui fait parti de cette catégorie de réseaux, nous avons considéré important de souligner ci-dessous

² <http://www.beanair.com/>

les principales caractéristiques des systèmes embarqués, celles qui les transforment dans une sous-classe de réseaux de capteurs sans fil classiques :

- ❖ La taille réduite du nœud capteur, ce qui peut faciliter un déploiement d'un nombre plus élevé d'éléments, étant donné le type d'alimentation;
- ❖ La redondance en cas de panne qui améliore elle aussi la densité du réseau ;
- ❖ Le traitement et l'agrégation des données qui diminuent la taille des paquets et en conséquence la consommation d'énergie, qui est un point sensible de ces systèmes, étant donné leur alimentation.

Pour résumer, le contexte dans lequel nous nous plaçons dans notre travail de recherche est, plus précisément, celui des réseaux embarqués pour l'instrumentation et la mesure. Mais, avant de rentrer dans les détails de notre application, nous allons nous intéresser dans les sections suivantes à des aspects concernant la mise en œuvre et les méthodes de déploiement, ainsi qu'à d'autres possibles applications.

I.1.2.1 Mise en œuvre

Toutes les caractéristiques identifiées auparavant vont avoir une influence sur le comportement global du réseau et vont faire de la mise en œuvre un travail difficile et unique pour chaque application donnée, parce que le « terrain », les contraintes et les besoins sont toujours spécifiques.

Ainsi, une première action consiste à établir un « état des lieux » avec les propriétés de l'environnement et choisir le type de capteur en fonction de ses caractéristiques physiques et techniques (jetables ou pas, avec ou sans mémoire, le type de batterie, la taille, le coût, etc.). Il faut tenir compte que, pour la plupart des applications, l'emplacement offre des ressources en énergie limitées, ce qui a un impact direct sur la façon de concevoir le système de communication (privilégier des technologies de transmission radio peu consommatrices en énergie, faire le choix d'un routage efficace, etc.).

L'architecture du réseau est un autre aspect important de la mise en œuvre, qui va être choisie en fonction du contexte (soit une topologie ad-hoc sans infrastructure, soit cellulaire avec infrastructure). Ce qui caractérise l'architecture de ces réseaux est une phase d'initialisation obligatoire, qui va permettre aux nœuds de découvrir ses voisins. Ils vont ainsi pouvoir interagir avec l'environnement physique, répondre à des requêtes distantes et transmettre les informations collectées. Nous rajoutons que, généralement, les liens établis sont asymétriques, ce qui peut poser des problèmes au niveau de précision de la synchronisation [6].

En ce qui concerne la bande passante (souvent limitée), la stratégie est d'utiliser des mécanismes de QoS (Quality of Service) adaptées pour chaque type d'application. En parallèle il faut assurer la fiabilité du système, donc éviter les erreurs de transmission, car les retransmissions vont contribuer à l'augmentation de délais. Mais, obtenir un bon compromis entre l'économie d'énergie et la bande passante et/ou les délais de transmission n'est pas une tâche aisée [7].

De plus, le coût de la mise en œuvre doit rester compatible avec l'application. Ça ne vise pas que le coût du matériel lui-même, mais aussi sa résistance aux conditions extérieures variées (pression, humidité, variations des températures, etc.), la facilité de déploiement et les ressources humaines impliquées.

I.1.2.2 Déploiement

Si nous analysons maintenait l'aspect « déploiement de nœuds capteurs » en soi, nous pouvons constater qu'il y a une grande diversité d'options pour accomplir cette lourde tâche.

Ainsi, il existe des architectures réseau déjà fixées dans lesquelles les nœuds capteurs peuvent être déployés d'une manière déterministe, ou bien des applications pour lesquelles le déploiement de nœuds capteurs est totalement aléatoire (par exemple les nœuds déployés depuis un avion [8]). En fonction de l'application, ce processus peut se réaliser pendant toute la durée de vie du réseau (de nouveaux nœuds capteurs sont intégrés au fur et à mesure dans l'environnement surveillé), ou bien se faire d'un seul coup à la création du système.

Le processus de déploiement sera différent aussi en fonction de la mobilité des nœuds capteurs, qui peut être due à l'attachement de nœuds à d'autres objets (dans les applications de logistique, par exemple).

Comme le temps et le coût d'un déploiement de réseau de capteurs ne sont pas négligeables, il y a de nombreuses applications pour lesquelles le déploiement aléatoire réalise le meilleur compromis. Mais, les options de maintenance dans de telles conditions de déploiement ne sont pas toujours évidentes. Ainsi, la question qui se pose est de savoir s'il y a la possibilité pratique d'effectuer de la maintenance sur ces capteurs.

Étroitement liée à la maintenance, les problèmes d'approvisionnement en énergie restent primordiaux. En fonction de la durée de la mission, l'approvisionnement en énergie peut être un problème, surtout quand l'entretien du matériel n'est pas possible et que les nœuds doivent travailler pendant des années.

I.1.2.3 Exemple d'applications des WSNs

Dû aux progrès des technologies de conception de capteurs, aujourd'hui il existe de nombreux domaines où les WSNs peuvent apporter leurs avantages : l'industriel, le médical, le militaire, l'agriculture, la sécurité, l'écologie, etc.

LE MILITAIRE

Un des premiers domaines d'applications pour les réseaux de capteurs a été le domaine militaire. Ainsi, le fort besoin pour le contrôle et la surveillance a transformé la technologie de WSN en objectif majeur [9], car nous le retrouvons :

- ❖ Dans la surveillance des bâtiments ou des périmètres ;
- ❖ Dans les missions d'espionnages ;
- ❖ Dans la détection des dangers comme les radiations, les agents chimiques, etc. ;
- ❖ Dans le contrôle et l'assistance pour des attaques terrestres ou aériennes ;
- ❖ Dans le suivi des troupes et des véhicules militaires.

LA SANTE

Dans le domaine de la santé, l'utilisation de WSN apporte beaucoup de bénéfices [82]. Les solutions envisagées sont multiples :

- ❖ Les soins post opératoires et intensifs (où les capteurs peuvent être placés plus facilement en faisant disparaître les câbles) ;

- ❖ La surveillance à long terme de patients (généralement pour les personnes âgées) ;
- ❖ L'administration automatique de médicaments ;
- ❖ Les systèmes de suivi des patients et médecins dans les hôpitaux, qui peuvent sauver des vies grâce à des alarmes intégrées.

L'INSTRUMENTATION ET LA MESURE

Un nœud capteur peut être chargé de rendre des rapports périodiques concernant des valeurs mesurées, et souvent ces rapports peuvent être déclenchés par un événement détecté. C'est le cas d'un réseau d'instrumentation et de mesures, où les informations détectées sont envoyées en mode continu à un nœud collecteur.

Par conséquent, les réseaux d'instrumentation et de mesures offrent une nouvelle génération de services, à un coût réduit et avec une augmentation de l'efficacité pour les applications industrielles. La technologie de détection et de contrôle des structures maintient donc un potentiel de progrès significatifs, non seulement en sciences et en génie, mais aussi dans le contrôle des infrastructures. Un exemple est celui des capteurs de pression fixés sur les ailes de l'avion [83].

AUTRES APPLICATIONS

D'autres applications des WSNs se sont fortement développées au long de ces dernières années, comme : les applications dans les usines de traitement des eaux usées [84], la détection des inondations [85], des musées interactifs [86], le suivi de l'habitat des oiseaux [87], etc.

Il existe aussi des applications avec des capacités de détection encore plus sophistiquées, comme par exemple des jouets qui peuvent avoir des capteurs tactiles ou de mouvement dans un endroit aménagé pour les enfants [81].

I.1.3 Le nœud capteur

La construction effective d'un réseau de capteurs sans fil nécessite le développement de nœuds qui doivent répondre à des exigences spécifiques aux applications, comme d'être : aussi petits que possible, avec un faible prix [1], efficaces en consommation d'énergie [74], équipés avec les ressources de calcul et de la mémoire nécessaires, ainsi qu'avec de moyens de communication adéquats.

I.1.3.1 *Composants matériels et logiciels*

Généralement, un nœud capteur est constitué d'un certain nombre spécifiques d'éléments (Figure I.2) :

- ❖ Un module *d'acquisition des données* pour la réception et la conversion des données analogiques en signaux numériques ;
- ❖ Un module de *traitement du signal* (DSP (Digital Signal Processors) + un conditionneur de signaux + un convertisseur) ;
- ❖ Une *unité de traitement des données* pilotée par un système d'exploitation (un processeur + une mémoire RAM (Random Access Memory) et ROM/EEPROM (Read Only Memory/ Electrically Erasable and Programmable ROM)) ;

- ❖ Un module de communication pour *l'émission/réception* ;
- ❖ Une *unité d'alimentation autonome*.

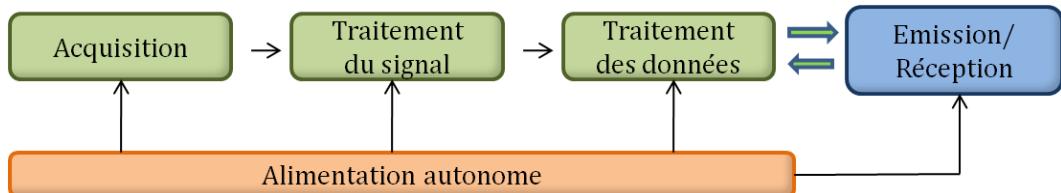


Figure I.2 Les principaux composants matériels d'un nœud capteur.

Chacune de ces unités doit opérer d'une manière équilibrée, en respectant le compromis entre une consommation d'énergie la plus petite possible et la nécessité de s'acquitter correctement de leurs tâches.

L'unité de traitement des données représente « le cœur » d'un nœud capteur sans fil. Son rôle est de recueillir et traiter les données provenant des capteurs, de décider quand et où les envoyer, de recevoir les données des autres nœuds et de décider sur le comportement de capteur en exécutant divers programmes. Une telle variété de tâches de traitement impose un compromis entre la flexibilité, les performances, l'efficacité énergétique et le coût de production.

L'ENVIRONNEMENT MATERIEL

Utiliser des processeurs à usage général comme ceux présentés sur un ordinateur fixe serait impossible à cause de leur consommation excessive en énergie. Ainsi, des processeurs spécifiquement utilisés dans les systèmes embarqués ont été conçus: *les microcontrôleurs*. Ils sont appropriés pour les réseaux de capteurs, car, entre autres, ils ont généralement la possibilité de réduire leur consommation d'énergie en se mettant en état de veille, ce qui signifie que seulement certaines parties du contrôleur seront actifs.

Un type particulier de processeur programmable est le *DSP* [73] ; il est adapté pour le traitement de grandes quantités de données vectorielles, comme dans les applications de traitement du signal et il présente aussi l'avantage d'une communication sans fil à large bande. Mais comme les réseaux de capteurs sans fil ont des exigences plus modestes en matière de communication et que les tâches de traitement du signal liées à la détection de données sont également peu compliquées, les avantages d'un *DSP* ne sont pas nécessaires dans un nœud capteur, ce qui explique l'effet qu'ils ne sont pas généralement utilisés.

Une autre option possible est d'utiliser un *FPGA* (Field-Programmable Gate Arrays) ou un *ASIC* (Application-Specific Integrated Circuit). Un *FPGA* peut être reprogrammé (ou plutôt reconfiguré) pour s'adapter à l'évolution d'un ensemble d'exigences, mais cela peut prendre du temps et de l'énergie. Un *ASIC* est un processeur spécialisé, conçu sur mesure pour une application donnée. Le type de compromis ici est la perte de flexibilité en échange d'une bonne performance et d'une efficacité énergétique nettement meilleure. D'autre part, où un microcontrôleur nécessite le développement de logiciels, les *ASIC* offrent les mêmes fonctionnalités matérielles, ce qui entraîne un développement matériel potentiellement plus coûteux.

Les principales caractéristiques d'un circuit *FPGA* sont :

- ❖ Une grande puissance de calcul ;
- ❖ Une adaptation spécifique avec l'algorithme à développer ;
- ❖ Une possibilité de reconfiguration ;
- ❖ Un temps de fabrication du circuit très réduit ;

- ❖ Un faible coût de production ;
- ❖ Un prototypage simple et rapide.

En ce qui concerne le choix du support de transmission pour la communication sans fil, le plus pertinent semble être le RF (Radio Frequency), car il correspond le mieux aux exigences des applications d'un WSN. Il peut offrir une couverture relativement longue, des débits élevés, des taux d'erreur acceptables avec une dépense énergétique raisonnable, sans avoir besoin d'une ligne de vue entre l'émetteur et le récepteur.

L'ENVIRONNEMENT LOGICIEL

L'élément indispensable pour la technologie matérielle de base est représenté par l'environnement d'exécution. Cet environnement doit prendre en charge des tâches simples, l'échange d'informations inter couches et la modularité pour permettre une maintenance efficace du système.

Concernant le système d'exploitation ou l'environnement d'exécution, ils doivent répondre à des besoins spécifiques pour les réseaux de capteurs. Ainsi, la nécessité d'une exécution à haut rendement énergétique demande des solutions pour la gestion de l'énergie, les composants externes – les capteurs, le modem radio, les horloges – qui doivent être manipulés facilement et efficacement, etc. Tout cela nécessite un modèle clair pour structurer la pile de protocoles, avec une priorité pour la gestion d'énergie, sans endommager les ressources système rares comme la mémoire ou le temps d'exécution.

En conclusion, nous pouvons supposer que le nœud capteur sans fil se compose de deux parties distinctes [76]: une partie qui peut détecter et signaler les événements avec une consommation faible, voire négligeable, et une autre partie qui effectue le traitement/calcul effectif et la communication, avec une consommation d'énergie non négligeable. Face à une grande variété de composants à choisir, avec leurs propres avantages et inconvénients, il n'est pas surprenant de conclure qu'il n'existe pas une solution « parfaite » pour réaliser un nœud capteur sans fil. Les exigences variées des applications vont demander différents compromis à respecter et différentes architectures à mettre en oeuvre.

I.1.3.2 Modules de nœuds capteurs existants

Nous rappelons encore une fois que, selon le scénario prévu dans l'application, les nœuds doivent satisfaire des exigences en ce qui concerne la vie de la batterie, la résistance mécanique en fonction de l'emplacement du nœud, la taille, etc.

À partir des années 1990, c'est la famille de nœuds « Mica motes » qui s'est développée, avec ses différentes versions (Mica, Mica2, Micaz, Mica2Dot) [76], [77], comme le fruit de collaboration entre l'Université de Californie à Berkeley et Intel Labs³ (cette solution est visible sur la Figure I.3).

³ <http://berkeley.intel-research.net/>



Figure I.3 Nœud capteur Mica2 Mote.

Le matériel est disponible dans le commerce via la société Crossbow⁴ en différents kits, ayant habituellement comme système d'exploitation le TinyOS. Le matériel dispose d'un microcontrôleur de la famille Atmel, un modem radio (généralement une TR 1000 de RFM) et différentes entités de sorties. Il a été construit pour une utilisation dans des environnements rudes, comme par exemple pour le suivi d'habitats d'oiseaux.

Le projet EYES⁵ (Energy efficient Sensor Networks) a donné naissance à un autre type de nœud capteur, équipé d'un microcontrôleur Texas Instrument MSP 430, un modem radio Infineon TDA 5250, où le modem radio envoie régulièrement des données concernant la puissance du signal mesuré au contrôleur. Le nœud est également équipé d'une interface USB, comme nous pouvons le visualiser dans la Figure I.4:



Figure I.4 Nœud capteur du projet EYES.

Un autre type de nœud capteur (Figure I.5) a été développé pour la plate-forme ScatterWeb [79] à Freie University, Berlin. Elle est constituée d'une multitude de nœuds relativement standards, basés sur un microcontrôleur de type MSP 430 et équipés d'une multitude d'entités pour l'interconnexion, avec des options qui impliquent aussi des connexions pour I2C ou pour des bus CAN.



Figure I.5 Nœud capteur de la plate-forme ScatterWeb.

Le nœud BTnode [78] présenté dans la Figure I.6, a été développé à l'ETH Zurich, et dispose d'un microcontrôleur Atmel ATmega 128L, (64 + 180) ko de RAM et 128 Ko de mémoire FLASH. Contrairement à la plupart des autres nœuds capteurs, il utilise sa propre technologie radio Bluetooth en combinaison avec une fonctionnalité Chipcon CC1000 entre 433 et 915 MHz.



Figure I.6 Nœud capteur BTnode.

⁴ <http://www.xbow.com>

⁵ <http://www.eyes.eu.org>

Les exemples que nous venons de donner représentent des solutions fiables pour des applications réelles, mais qui sont soumis à des améliorations dû à l'évolution de la recherche. Cette évolution perpétuelle impose comme nécessité, le développement pour de futurs nœuds capteurs avec des capacités émergentes en matière de communication, de calcul, de stockage et de meilleures stratégies pour une consommation minimale d'énergie.

I.1.3.3 La pile protocolaire

Le principal objectif face à une application de réseau de capteurs sans fil est de pouvoir fournir une solution efficace, qui :

- ❖ Soit basée sur des normes de réseau sans fil ;
- ❖ Peut gérer des faibles/moyens débits de données, avec une faible consommation en énergie ;
- ❖ Peut supporter la mobilité (capacité d'auto-organisation) ;
- ❖ Peut intégrer une gestion des tâches, tout en restant fiable.

La Figure I.7 représente une vision possible de la pile protocolaire pour les réseaux de capteurs sans fil en général.

Nous pouvons donc constater que l'ensemble des protocoles doivent assurer la communication et en même temps s'acquitter de la gestion globale du système. Pour traiter ce deuxième aspect (qui est spécifique pour les réseaux de capteurs sans fil), il existe plusieurs solutions :

- ❖ Confier les différents plans de gestion à un certain niveau de communication ;
- ❖ Faire de la gestion en impliquant plusieurs niveaux de communication en même temps ;
- ❖ Combiner les deux possibilités.

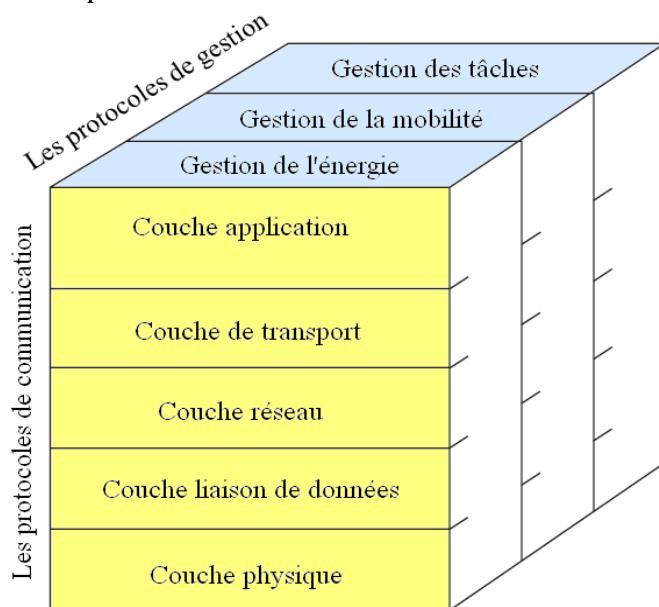


Figure I.7 Modèle de la pile protocolaire pour WSNs.

Dans tous les cas, le « dénominateur » commun de toutes ces méthodes est ce que nous appelons « cross-layering ».

LA COUCHE PHYSIQUE

Le niveau physique est responsable de la connectivité et de la couverture du réseau. Son objectif est ici d'interconnecter un grand nombre de capteurs qui sont dispersés et/ou mobiles, d'une manière rentable et fiable. A travers ses options de sélection de la fréquence de communication, de détection et traitement du signal, de localisation, cette couche peut servir comme base pour la gestion de la mobilité, par exemple.

LA COUCHE LIAISON DE DONNEES (MAC)

La couche liaison de données ou MAC, se concentre sur l'aspect contrôle d'erreurs et offre des mécanismes d'auto-organisation du réseau et de communication multi-sauts. Il ne faut pas oublier que le choix de la méthode d'accès au medium est un des facteurs déterminants dans la performance d'un réseau de capteur sans fil. Pour résoudre le problème de l'accès partagé, plusieurs stratégies ont été proposées : l'allocation fixe, l'allocation à la demande et l'accès aléatoire, qui intègrent aussi des modèles pour la gestion dynamique de l'énergie.

Les protocoles typiques qui font partie de la première catégorie, la méthode d'allocation fixe [92], sont le FDMA (Frequency-Division Multiple Access), le TDMA (Time-Division Multiple Access), et le CDMA (Code-Division Multiple Access).

Les protocoles qui utilisent les méthodes d'allocation à la demande vont ignorer les nœuds actifs et vont considérer seulement les nœuds prêts à transmettre. L'inconvénient est que la nécessité de demander l'accès au canal peut retarder la transmission. Ces protocoles peuvent être classés en deux catégories : centralisée (comme par exemple le schéma de type « polling » [93]) ou distribuée (des schémas de réservation ou à base de jeton, comme le protocole ALOHA [94]).

Si la source de trafic envoie des données en rafale, le régime d'allocation fixe peut être inefficace, car l'inactivité des nœuds entraîne une perte de la bande passante déjà allouée. C'est ici que les stratégies d'allocation aléatoire ont tenté de remédier à cette lacune, en éliminant la pré-allocation de la bande passante à tous les nœuds du réseau. Le principal risque est l'apparition des collisions, et pour faire face à ce problème les protocoles doivent inclure des mécanismes de détection des collisions et un calendrier pour les ultérieures retransmissions. Dans ce registre, les schémas les plus connus sont [34]: le CSMA (Carrier-Sens Multiple Access), le CSMA/CD (Carrier-Sens Multiple Access with Collision Detection) et le CSMA/CA (Carrier-Sens Multiple Access with Collision Avoidance).

LA COUCHE RESEAU

Le protocole réseau est responsable de l'adressage des éléments. Pour un réseau de capteurs, il doit garantir un adressage par attributs et une localisation précise, pour pouvoir appliquer ensuite des stratégies de routage à basse consommation d'énergie (trouver la route avec le maximum d'énergie disponible, avec un minimum de sauts, etc.).

Dans la suite, nous présentons quelques exemples de tels protocoles existants dans la littérature.

Ainsi, le SMECN (Small Minimum Energy Communication Network) utilise les informations de localisation pour déterminer la consommation d'énergie entre deux nœuds donnés, pour un routage efficace des données de point de vue énergétique [89].

SPIN (Sensor Protocol for Information via Negotiation) est un protocole de diffusion de données qui fonctionne sur le principe de diffuser efficacement les informations recueillies par des capteurs individuels à tous les nœuds du réseau [88].

Un protocole orienté QoS est le SAR (Sequential Assignment Routing) : il exploite les problèmes de délais de bout en bout pour la mise en place des chemins optimaux dans le réseau de capteurs [90].

LA COUCHE TRANSPORT

Généralement, les protocoles de transport apportent leur utilité dans les réseaux prévus pour être connectés à d'autres réseaux, comme l'Internet. Pour le cas des WSNs le niveau de communication transport peut être nécessaire pour assurer une transmission totalement fiable et permettre la reconfiguration des capteurs ou la réassignation de tâches dans le réseau. Un protocole évolutif dans ce sens est le PFSQ (Pump slowly, Fetch quickly) qui répond aux besoins des nouvelles applications dans les réseaux de capteurs [91].

LA COUCHE APPLICATION

En ce qui concerne la couche application, elle apporte sa contribution dans les réseaux de capteurs sans fil en satisfaisant les différents exigences du client, à travers des collaborations avec les autres couches de la pile protocolaire, dans le but de pouvoir proposer des services pour la gestion de l'énergie, la synchronisation, etc.

I.2 Principales caractéristiques d'un WSN

Les applications avec des réseaux de capteurs sans fil sont très différentes les unes des autres, ce qui représente un défi pour la recherche, car les problèmes techniques qui peuvent apparaître sont nombreux. En conséquence, chaque application va bénéficier d'une solution unique, qui ne fournira pas une conception généralisée pour ce type de systèmes.

Nous avons constaté à travers les exemples précédents, que dans de nombreuses applications avec des WSNs, les nœuds ne peuvent pas être facilement connectés à une alimentation par câble. Donc c'est plutôt une alimentation par batterie qui s'impose. Dans une telle situation, le besoin pour une consommation efficace en énergie est un aspect très important pour prolonger la durée de vie du système.

Pour d'autres applications, l'alimentation n'est pas un problème, mais ça peut être la précision des résultats fournis qui devient plus importante (i.e. la précision de la synchronisation).

Un autre aspect aussi important correspond à la taille et au coût du nœud, qui doivent rester convenables pour chaque type d'application.

Ainsi, pour concevoir et mettre en œuvre des réseaux de capteurs sans fil, il faut avoir une bonne compréhension de l'application et des exigences qui en découlent. En même temps, pour choisir ou concevoir la solution la mieux adaptée, il faut avoir une bonne connaissance des technologies et concepts existants et largement utilisés dans l'univers des réseaux de communication. Pour cela, dans la section suivante nous proposons une comparaison entre les WSNs et les réseaux traditionnels, ainsi qu'une analyse des aspects qui les différencient ou qui les approchent les uns des autres.

I.2.1 Les réseaux traditionnels vs les WSNs

I.2.1.1 Typologie des réseaux sans fil

LE RESEAU AD HOC

Un exemple classique de système est le *réseau de communication ad hoc*. Ce type de réseau a été créé dans le but de répondre rapidement à des besoins spécifiques, comme dans l'exemple d'un réseau local. Dans un tel contexte, l'aspect d'auto-configuration peut être crucial si le réseau a été prévu de fonctionner sans spécification de gestion ou de configuration, aspect que nous retrouvons aussi dans les réseaux de capteurs sans fil.

Cependant, la notion de MANET (Mobile Ad Hoc Network) est associée à la communication sans fil et spécifiquement à la communication multi-sauts sans fil (Figure I.8). Elle indique aussi le caractère mobile pour les nœuds participants.

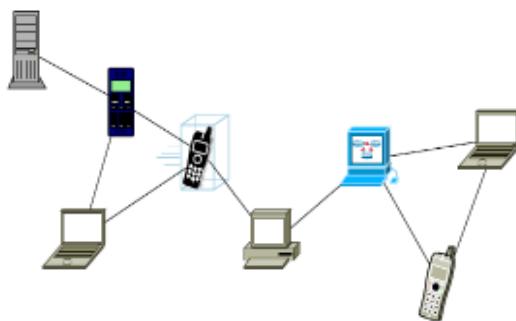


Figure I.8 Réseau MANET⁶.

Des exemples de tels réseaux sont les opérations de secours (sauveteurs, pompiers qui doivent communiquer les uns avec les autres) ou des réseaux dans des endroits difficilement accessibles (les grands chantiers de construction par exemple). Dans ces cas, les éléments individuels forment un réseau qui relaie les paquets entre tous les nœuds, permettant une plus grande couverture géographique, ce qui ne serait possible avec une communication directe expéditeur - récepteur. Deux grands défis dans un MANET sont la réorganisation du réseau lorsque les nœuds se déplacent et le traitement des problèmes de la portée limitée de la communication sans fil [10], [11]. Ces problèmes généraux sont présents aussi dans les réseaux de capteurs sans fil.

Mais, comme les applications visées sont différentes, les équipements utilisés ne seront pas les mêmes. Ainsi, dans un MANET, le terminal peut être assez puissant (un ordinateur portable ou un PDA) avec une batterie suffisante (par exemple les applications qui doivent assurer la communication vocale entre deux nœuds éloignés, ou l'accès à une infrastructure à distance comme un serveur Web), ce qui n'est pas le cas dans un réseau de capteurs.

Il est donc impossible qu'il y ait une solution pour tous les différents WSNs, car les réseaux de capteurs sont envisageables, entre autres, avec des densités très différentes (de très clairsemée à un déploiement très dense). Cette diversité nécessitera des adaptations ou des améliorations dans toute la pile des protocoles, diversité qui jusqu'à présent, n'est pas aussi grande dans les MANETs.

⁶ <http://www.laria.u-picardie.fr/~ids-laria/research.html>

En conclusion, nous pouvons constater qu'il y a des points communs entre ces deux concepts de réseaux. Par contre, le fait que les réseaux de capteurs puissent supporter des applications très différentes, interagir avec l'environnement physique et gérer les différents compromis qui peuvent apparaître, nous permet à considérer les WSNs comme un concept différent des MANETs.

LE BUS DE TERRAIN

Un autre type de réseau très connu dans la littérature correspond au *bus de terrain*. C'est un réseau spécialement conçu pour un fonctionnement sous des contraintes temps réel dures et le plus souvent avec la tolérance intégrée aux pannes. Il est destiné à être utilisé principalement dans des applications de contrôle (Figure I.9).

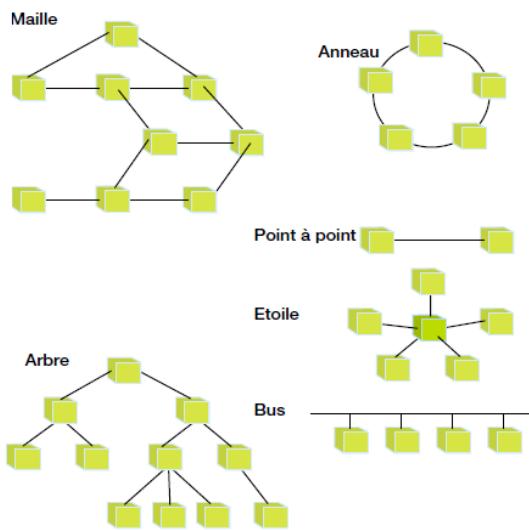


Figure I.9 Exemple de réseau bus de terrain⁷.

Un exemple de réseau embarqué destiné aux applications des voitures est le bus CAN [12], [13]. En raison d'exigences strictes en temps réel, ces réseaux sont généralement câblés et seules les couches physique, liaison de données et application du modèle ISO-OSI sont utilisées.

Les bus de terrain ont également pour objectif de faire face à l'environnement physique (comme la détection et le contrôle des données), ce qui les rapprochent des réseaux de capteurs sans fil.

Mais il y a aussi de grandes différences qui les éloignent, comme : l'adaptation des compromis que les réseaux de capteurs sont disposés à faire (par rapport à l'efficacité énergétique, par exemple) qui est un concept peu présent dans la littérature de bus de terrain; spécifiquement, les bus de terrain ne font aucune tentative pour économiser l'énergie, due à leur alimentation en continue.

I.2.1.2 *Les principales normes de communication sans fil*

Après avoir vu un certain nombre des concepts de réseaux de communication sans fil, nous nous proposons de présenter dans la suite du chapitre les normes de

⁷ http://www.ac-grenoble.fr/ecoletreprise/CRGE/ctressources/conf_schneider_autres/Reseau/GT_RESEAUX.pdf

communications existantes, qui pourront être potentiellement utilisées dans des réseaux de capteurs sans fil.

Commençons par la norme IEEE 802.15.4 [20], qui utilise le principe de communication CSMA/CA et qui a l'avantage d'une faible consommation en énergie. Parfois, cette norme est confondue avec la technologie ZigBee⁸, qui est une nouvelle norme utilisant les services offerts par l'IEEE 802.15.4. Ce qu'apporte en plus ZigBee est une restructuration du réseau (en étoile, en réseaux maillés, en réseaux « cluster-tree »), ainsi que des services de sécurité, de niveau applicatif, etc.

Les applications visées par l'IEEE 802.15.4 sont du domaine des réseaux de capteurs sans fil, la domotique, les réseaux domestiques, des dispositifs de connexion à un PC, la sécurité, etc. Ainsi, une nouvelle catégorie de réseaux est née (Figure I.10), le LR-WPAN (Low Rate - Wireless Personal Area Network), qui est caractérisée par un faible débit et une faible consommation en énergie, avec des topologies pair à pair ou en étoile.

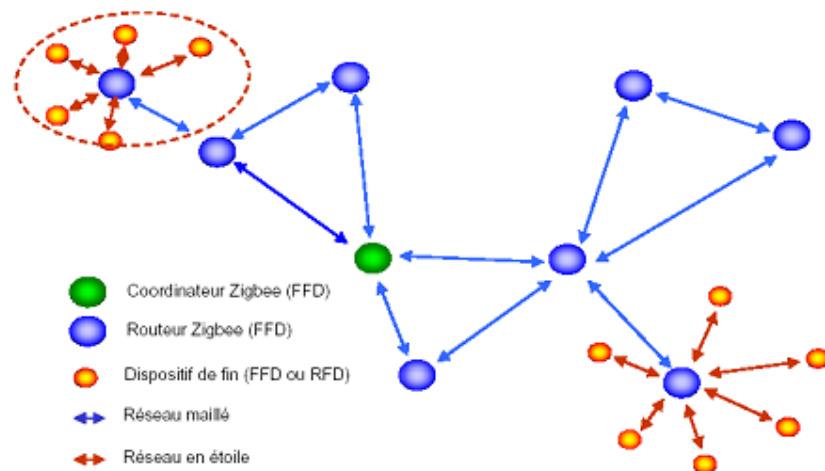


Figure I.10 Topologies LR-WPAN⁹.

L'UWB (Ultra Wide Band) est une génération de technologie de communication sans fil de courte distance qui consomme peu d'énergie et qui permet le transfert de données à haut débit. Elle a l'avantage et la mobilité de la communication sans fil qui assure l'interconnexion à une grande vitesse entre les dispositifs numériques, sa particularité étant de transmettre des successions d'impulsions très courtes, entre 10 et 1000 ps.

Une question à se poser est de savoir pourquoi d'autres normes avec un succès commercial comme IEEE 802.11 et Bluetooth ne sont pas les premiers choix dans les réseaux de capteurs sans fil. Nous essayons de trouver la réponse en regardant plus en détails ces technologies.

Ainsi, la technologie Bluetooth est conçue comme un réseau sans fil WPAN (Wireless Personal Area Network) avec une application qui est très réponduée: la connexion de périphériques à un ordinateur personnel [23]. Mais, cette technologie a été déjà utilisée comme un moyen de communication pour le prototypage des applications sans fil de réseau de capteurs [21].

⁸ <http://www.zigbee.org/>.

⁹ <http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>

Les réseaux qui utilisent la technologie Bluetooth, sont habituellement organisés en « piconet » et « scatternet » avec un maître et jusqu'à sept nœuds esclaves actifs ; ainsi le maître choisit la séquence de saut où les esclaves doivent le suivre. Un exemple est présenté dans la Figure I.11, où le réseau « scatternet » est constitué de 3 « piconet ».

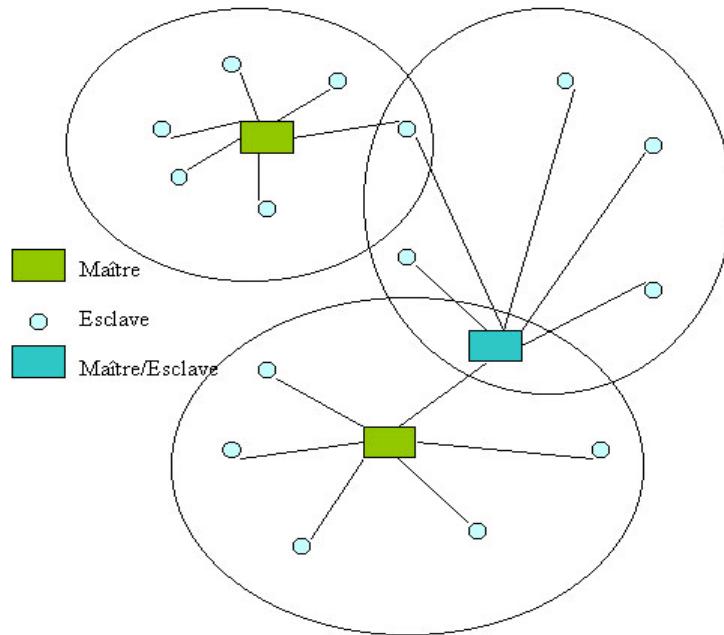


Figure I.11 Réseau scatternet Bluetooth¹⁰.

Nous notons les deux inconvénients majeurs de cette technologie :

- ❖ Le besoin d'avoir constamment un nœud maître qui dépense beaucoup d'énergie sur les sondages de ses esclaves et le nombre relativement limité d'esclaves actif par « pico réseau ». Ceci n'est pas compatible avec le cas de réseaux de capteurs sans fil denses, où un grand nombre de nœuds maître seraient nécessaires.
- ❖ En outre, il est nécessaire que chaque nœud soit capable de prendre le rôle de maître ou d'esclave, ce qui porte une complexité considérable. Aussi, la fréquence rapide de saut entre les opérations nécessite une synchronisation étroite entre les nœuds d'un tel réseau.

En ce qui concerne la norme IEEE 802.11, elle présente certaines fonctionnalités pour faire économiser l'énergie [27]. Mais, comme en général un tel système est caractérisé par des débits aussi importants où les émetteurs-récepteurs requièrent une grande quantité d'énergie, cela n'est pas adapté dans les applications à faible débit comme les réseaux de capteurs.

L'IEEE 802.11 a été conçue à la fois pour des infrastructures de réseau ad hoc et, en général, est destinée à laisser un certain nombre d'utilisateurs à partager le canal de communication, fonctionnement qui ne correspond pas aux objectifs des réseaux de capteurs sans fil.

Ainsi, les technologies discutées auparavant présentent un certain nombre des avantages et des inconvénients (l'IEEE 802.11 a une portée et une bande passante plus importante, la faible consommation d'énergie de ZigBee représente un atout non négligeable, etc.) qui sont résumés dans le Tableau I.1:

¹⁰ <http://tp.bluetooth.free.fr/archi.html>

| Norme | Portée | Fréquence | Débit maximal | Puissance (en marche) | Informations complémentaires |
|------------------------|-------------|-------------------|---------------|-----------------------|--|
| IEEE 802.11 (1997) | 500 m | 2.4 – 5 GHz | 54 Mb/s | 350 mA | Norme bien maîtrisée, très répandue, avec des performances limitées, peu coûteuse. |
| WiMedia (2002) | 1 Km | 3.1 – 10.6 GHz | 480 Mb/s | 35 mA | Basse consommation, avantage rapport prix/performances. |
| ZigBee (2004) | 0 m -1.5 Km | 2.4 GHz – 868 MHz | 250 Mb/s | 50 mA | Basse consommation, gestion sécurisée et fiable, peut lier des milliers d'appareils dans le même réseau. |
| Bluetooth (début 2010) | 100 m | 2.4 GHz | 1 Mb/s | 100 mA | Connexion immédiate, configuration simple et rapide. |

Tableau I.1 Caractéristiques pour des technologies de communications sans fil.

Ainsi, toutes ces normes ont été conçues dans le but d'économiser l'énergie, réduire les délais ou assurer un haut débit.

Nous pouvons conclure qu'il n'y a pas de « meilleur protocole MAC » ; le bon choix dépend de chaque application, des modèles de batteries prévues, du déploiement et du comportement du matériel sous-jacent d'un point de vue consommation d'énergie (la consommation relative de la transmission à la réception, pendant les temps de réveil et en mode veille ainsi que les coûts de calcul spécifiques pour l'exécution du protocole MAC).

En ce qui concerne l'aspect lié à la consommation d'énergie dans les réseaux de capteurs sans fil, il se relève très important, raison pour lequel nous avons décidé d'aborder ce sujet plus en détail dans la section I.2.2.

I.2.1.3 Protocoles de communication pour les WSNs

Nous nous intéressons maintenant à savoir comment concevoir des protocoles/interfaces de communication appropriés aux WSNs. Pour cela, la suite de notre étude est dédiée à des aspects concernant l'architecture réseau de ces systèmes, dans le but de résoudre les problèmes de communication.

La richesse des options qu'une interface générale dédiée aux réseaux de capteurs aurait à offrir est vaste. En regardant les questions/problèmes soulevés jusqu'ici, trois d'entre eux se mettent en évidence :

- ❖ La communication qui est centrée sur les données ;
- ❖ Le besoin d'une bonne gestion de l'énergie ;
- ❖ Le compromis entre une bonne précision de la synchronisation et la consommation en énergie.

En conséquence, concevoir des WSNs qui sont très différents les uns des autres en utilisant des concepts et des technologies existants aujourd'hui, est un défi loin d'être réalisé. Il n'est donc pas si surprenant qu'il y ait eu relativement peu de travail sur une approche systématique de service pour les interfaces de réseaux de capteurs sans fil.

Une tentative dans ce sens a été entreprise par Sgroi et al. [14], qui se base sur le principe d'une interface client/serveur pour organiser une «gestion de requêtes» et une «interface de commande», embellies par d'autres ensembles de paramètres. Mais, il reste particulièrement difficile de savoir comment étendre les fonctionnalités de gestion en réseau en fonction de chaque API (Application Programming Interface), comment contrôler la consommation d'énergie, ou comment sélectionner à partir d'une seule commande plusieurs composants pour exécuter une tâche donnée (par exemple, sélectionner un des protocoles de routage disponibles pour l'application).

Un autre exemple d'application très spécifique des demandes de services est EnviroTrack [1], qui a comme but principal le suivi d'objets mobiles. Cette application permet de définir des «contextes» pour certaines tâches de suivi, qui ont des fonctions d'activation et de «rapport» des objets.

Donc, nous constatons que les interfaces de communication pour WSNs sont d'une très grande complexité à cause des exigences des applications. Pour mieux répondre à ces exigences, cela nécessite une meilleure compréhension de la pile protocolaire, ou de mettre en balance les efforts nécessaires pour l'optimisation des performances, lorsque nous utilisons une interface de service prédéfinie.

PROTOCOLES DE NIVEAU MAC

Le protocole MAC est la première couche au-dessus de la couche Physique (PHY) et, par conséquent, les protocoles MAC sont fortement influencés par les propriétés de cette couche. La tâche fondamentale de tout protocole MAC est de réglementer l'accès d'un nombre de nœuds à un support partagé, de manière à ce que certaines exigences de performance dépendantes de l'application soient remplies. Habituellement, les exigences de performances les plus importantes pour les protocoles MAC sont :

- ❖ L'efficacité en terme de débit ;
- ❖ La stabilité du système ;
- ❖ Un faible délai d'accès au média (le temps entre l'arrivée du paquet et la première tentative réussite de le transmettre) ;
- ❖ Un faible délai de transmission (le temps entre l'arrivée de paquets et la réussite de la transmission).

Dans la littérature, nous trouvons des nombreuses solutions pour améliorer le fonctionnement de la norme 802.11, comme par exemple l'adapter au contexte des antennes directives [22]. Dans cette catégorie, nous pouvons citer les protocoles comme T-MAC [24], B-MAC [25], ou bien des propositions hybrides [26].

Dans les réseaux des capteurs sans fil, la question de l'efficacité énergétique est de plus en plus considérée au niveau des protocoles MAC. L'une des principales approches pour économiser l'énergie consiste à mettre les nœuds en état de sommeil le plus de temps possible. L'idée est de passer un maximum de temps dans l'état de repos et de réduire au maximum les activités de communication d'un nœud capteur [95].

Pour d'autres protocoles, par exemple le S-MAC (Sensor - MAC) [15], il y a aussi une phase de réveil périodique, mais les nœuds peuvent à la fois transmettre et recevoir au cours de cette phase de réveil. Lorsque les nœuds se trouvent dans leur phase de réveil en même temps, il n'est pas nécessaire pour un nœud qui souhaite transmettre un paquet, d'être réveillé en dehors de ces phases de rendez-vous avec son récepteur.

D'autres classes importantes de protocoles MAC sont axées sur le problème des collisions. Dans ce contexte, les données à transmettre vers un possible nœud récepteur peuvent être interceptées par un de ses voisins. Le risque de se retrouver avec plusieurs nœuds voisins qui attendent le canal en même temps, est de provoquer des collisions qui vont augmenter la consommation en énergie. Il y a deux protocoles importants faisant partie de cette catégorie [17] : ALOHA et CSMA, auxquels se rajoutent des mécanismes pour résoudre le problème du terminal caché. Adireddy et Tong [16] ont pris en compte des informations théoriques sur les réseaux de capteurs sans fil, et plus précisément, ils ont étudié une variation d'ALOHA qui utilise des informations sur l'état du canal.

Nous n'oubliions pas les protocoles d'accès au canal à base de calendrier, pour lesquels les utilisateurs peuvent partager le medium de communication via différentes méthodes [18], telles que TDMA, FDMA, CDMA, ou SDMA. L'avantage fondamental de ces protocoles est que les horaires de transmission sont calculés de telle sorte qu'il n'y a pas de collision à la réception et donc des mécanismes spéciaux pour éviter des situations telles que « le nœud caché » sont inutiles.

L'algorithme du TDMA [19] attribue explicitement des périodes de transmission et de réception pour les nœuds, en les laissant dormir tout le reste du temps. En contrepartie, il est difficile pour un nœud de céder ses slots de temps à ses voisins. Un autre inconvénient est que le calendrier d'un nœud (et éventuellement celui de ses voisins) peut nécessiter une quantité importante de mémoire, qui est une ressource rare dans plusieurs applications. Enfin, l'attribution des horaires TDMA distribués sans conflit est un problème difficile en lui-même.

Nous pouvons constater que le problème lié à la consommation d'énergie est un aspect souvent traité dans la conception des technologies de communication sans fil. Une fois de plus pour les WSNs, la gestion d'énergie est un besoin indispensable qui influence directement la durée de vie du système. Par conséquent, dans la section suivante, nous avons analysé de plus près ce paramètre avec les possibles méthodes d'optimisations trouvées dans la littérature.

I.2.2 La limitation en énergie pour un WSN

Concevoir des réseaux de capteurs sans fil est devenu possible grâce à certains progrès décisifs dans les technologies utilisées, et en particulier la miniaturisation du matériel. Ainsi, les dimensions microscopiques des éléments ont également fait baisser la consommation d'énergie pour les composants de base d'un nœud capteur, à un niveau tel que la construction de réseaux de capteurs peut être envisagée. Ceci est particulièrement pertinent pour les microcontrôleurs et les mémoires, mais aussi pour les modules radio dont la communication sans fil est devenue beaucoup plus éco-énergétique. La réduction de taille des puces permet aussi un déploiement redondant de nœuds, amenant par conséquent une réduction du coût de production.

Donc, une des préoccupations de base concernant le nœud capteur est sa consommation en énergie. Cela nécessite dans la plupart des applications des batteries haute capacité, qui ont un taux négligeable d'autodécharge. Idéalement, un nœud capteur a également un dispositif de récupération d'énergie, comme des cellules solaires¹¹ (Figure I.12 (a)), ou des dispositifs de récupération d'énergie thermoélectrique¹² (Figure I.12 (b)). Les deux problématiques, la batterie et la récupération de l'énergie représentent aujourd'hui des importants sujets de recherche.



Figure I.12 Différents dispositifs de récupération d'énergie.

L'avantage de ne pas être obligé de raccorder l'équipement à une source électrique est donc la liberté de mouvement dont bénéficient les nœuds capteurs sans fil, ce qui donne une grande souplesse en terme de mobilité au réseau. En contre partie, l'alimentation en énergie devient l'élément crucial du système. Il existe essentiellement deux aspects à traiter : d'une part, stocker l'énergie et pouvoir la fournir dans la forme requise, et d'autre part, tenter de reconstituer l'énergie au fur et à mesure à partir d'une source externe.

En conséquence, la source d'énergie d'un nœud capteur peut être une batterie non rechargeable (des piles primaires), ou rechargeables si un périphérique de stockage est présent sur le nœud. Comme la récupération d'énergie est devenue un important sujet de recherche [32], la réalisation des batteries est devenue une véritable industrie. Sous une forme ou une autre, les batteries représentent une accumulation électrochimique d'énergie. Les produits chimiques constituent le principal facteur de la technologie de fabrication des batteries. Ils doivent avoir une capacité de stockage élevée mais un faible poids, un faible volume, ainsi qu'un prix bas.

Généralement, les batteries ont une capacité faible et la recharge par récupération de l'énergie est complexe et pas suffisante, ce qui impose un contrôle étroit de la consommation énergétique d'un nœud capteur. Les principaux consommateurs d'énergie sont le microcontrôleur, la partie radio, dans une certaine mesure la mémoire, et, selon leur type, les capteurs.

La contribution la plus importante pour la réduction de la consommation énergétique de ces composants provient du niveau puce électronique et des technologies bas niveau. Mais, les avantages acquis par de tels composants peuvent facilement être perdus lorsque les composants sont mal exploités ou mal adaptés au contexte de l'application.

Comme pour certaines applications le nœud capteur n'est pas en activité tout le temps, il est par conséquent préférable de le désactiver. Éteindre complètement un nœud n'est pas possible, donc nous pouvons plutôt adapter son état de fonctionnement aux tâches à accomplir. Ainsi, l'introduction de plusieurs états de fonctionnement pour le

¹¹ <http://www.presence-pc.com/tests/PC-energie-solaire-22698/4/>

¹² <http://www.industrie.com/it/energie/une-alimentation-autonome-pour-capteurs-sans-fil.9840>

microcontrôleur est une technique majeure pour le nœud capteur sans fil à haut rendement énergétique [33].

Similaires aux états d'un microcontrôleur, l'émetteur et le récepteur radio peuvent fonctionner dans différents modes. Plus précisément, ils sont mis sous tension ou hors tension. Pour mieux comprendre le comportement de la consommation d'énergie pour les émetteurs-récepteurs radios et leur impact sur la conception des protocoles, des modèles de consommation d'énergie par bit ont été proposés et développés [34].

En faisant une analyse de la consommation de l'énergie des microcontrôleurs et des émetteurs-récepteurs radios [35], une question évidente se pose : quelle est la solution la plus optimale pour utiliser les ressources énergétiques d'un nœud capteur ? Autrement dit, est-il préférable de privilégier l'envoi des données ou plutôt de faire des calculs au niveau logiciel.

Dans [36], il est montré assez clairement que la communication est une tâche beaucoup plus « chère » que le calcul. Pourtant, l'énergie dépensée dans les calculs ne peut pas être simplement ignorée ; elle reste cependant moins importante que l'énergie consommée par la communication, mais toujours sensible dans la consommation globale. Cette observation est à la base d'un certain nombre d'approches et de décisions de conception d'architecture de réseau de capteurs sans fil.

L'idée est donc d'investir le plus possible dans le calcul informatique du réseau afin de gagner sur les dépenses énergétiques de la communication.

Nous avons parlé de l'influence exercée par « l'absence du fil » dans les WSNs, sur la consommation totale d'énergie. Nous rappelons que les technologies sans fil présente également des avantages, parmi lesquels la mobilité, que nous avons l'intention de la traiter dans la section suivante.

I.2.3 La mobilité

La mobilité est un aspect caractéristique pour les réseaux de capteurs sans fil qui n'est pas forcément exploitée par toutes les applications, car un certain nombre de WSN restent fixe. Donc, en fonction des besoins, nous pouvons classifier les WSNs en réseaux statiques ou mobiles.

Dans un réseau de capteurs statiques, les éléments déployés ainsi que l'observateur final (ou la station de base) n'ont pas besoin d'être considérés comme mobiles. Dans cette situation, pour pouvoir assurer la meilleure couverture de la zone d'intérêt, la phase de déploiement exige une étape de mise en place des infrastructures de communication adaptées pour chaque contexte donné.

En ce qui concerne le réseau de capteurs mobile, tous les éléments deviendront mobiles. Plus exactement, pour un nœud qui se déplace, sa zone d'intérêt changera, ce qui va entraîner aussi une « mobilité » de l'observateur par rapport à son ancienne position. Dans ce cas là, pour optimiser la communication, la station de base doit calculer pour un seul nœud plusieurs routes, parmi lesquelles elle pourra choisir celle qui sera la plus adaptée.

Des exemples d'application avec des capteurs mobiles sont les suivis des animaux dans leur déplacement quotidien, ou bien des missions militaires où les équipements des soldats peuvent être dotés de capteurs de surveillances.

Pour résumé, le grand avantage de cette mobilité est qu'elle permet au système de s'adapter à l'environnement, de s'auto-organiser, et donc de rendre le résultat final plus efficace. En contrepartie, la mobilité entraîne des problèmes dûs à la topologie dynamique et au routage, ainsi que de localisation.

Après avoir identifié les principales caractéristiques pour un WSN, nous analyserons dans la section suivante les possibles méthodes d'optimisations qui pourront nous permettre d'améliorer les paramètres de performances d'un réseau de capteurs sans fil. Cela va nous servir de base dans l'étape de conception et réalisation de notre système de communication sans fil.

I.3 Optimisation et paramètres de performances

I.3.1 L'optimisation à travers les principes du cross-layer

Habituellement, une architecture de communication bien adaptée à l'application favorise la longévité du système. Même si le médium sans fil est fondamentalement différent de celui filaire, une architecture classique en couches de communication est un moyen raisonnable d'exploiter les réseaux sans fil, qui restent optimal jusqu'à un certain point. Ainsi, les travaux d'optimisation sont devenus aujourd'hui une préoccupation importante pour la recherche. Une des méthodes les plus utilisées pour optimiser les systèmes de communications est le cross-layer (ou l'architecture inter-couche).

Le principe du cross-layer est une approche coopérative du fonctionnement des systèmes, et consiste à échanger des informations entre les différentes couches de communication en fournissant des informations à partir des couches inférieures, tout en gardant une générativité maximale. Ainsi les nouvelles architectures ont l'avantage de permettre un meilleur contrôle du médium de communication.

Autrement dit, les couches qui adressent des services dans une approche cross-layer peuvent être adjacentes ou pas (comme nous pouvons voir dans la Figure I.13, où les niveaux applicatif et physique par exemple, peuvent changer des informations), ce qui impose de bien définir les interactions entre ces couches non adjacentes.

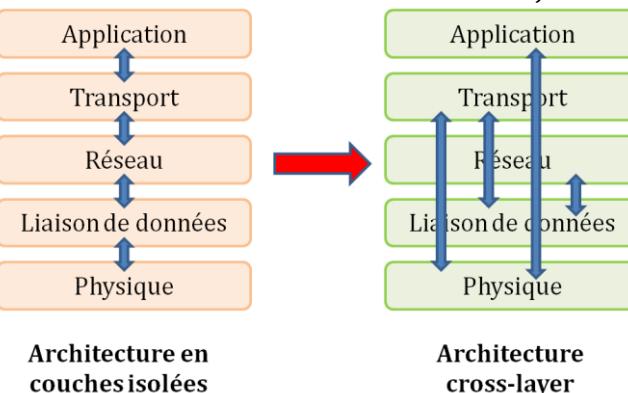


Figure I.13 Exemple d'architectures cross-layer générique.

Cependant, la nécessité d'améliorer le système en intégrant une adaptation inter-couche ne peut pas être ignorée, mais une telle approche peut également avoir des conséquences inattendues (comme par exemple, une très grande complexité), ce qui justifie l'effort de prendre du recul et d'examiner globalement le choix d'une telle architecture [28].

Les réseaux sans fil permettent aujourd’hui le développement de nouvelles applications, mais posent aussi des défis techniques importants. Dans [29], les auteurs montrent qu’une architecture de type cross-layer est indispensable pour répondre aux exigences des applications émergentes, en particulier quand l’énergie est une ressource limitée. Les plus grands défis de conception d’un réseau ad hoc sans fil par exemple, sont le manque de contrôle centralisé, une capacité limitée pour les nœuds, la variabilité des liens et la topologie du réseau. Les solutions pour relever ces défis sont des protocoles de contrôle intelligents, la redondance des nœuds, leur coordination, et l’adaptabilité à chaque couche de la pile de protocoles pour compenser la variabilité.

Dans ces conditions, une étude sur les aspects de mise en œuvre des couches de communication pour les réseaux de capteurs sans fil est nécessaire. Il faudra donc concevoir de nouveaux protocoles parce que les réseaux de capteurs sans fil nécessitent des fonctionnalités spécifiques de routage. En outre, des modules d’assistance tels que la synchronisation des données, le contrôle de la topologie, ou la localisation peuvent être nécessaires.

La conception de couches interconnectées reste particulièrement importante en raison des contraintes énergétiques, puisque l’énergie consommée à travers toute la pile de protocoles doit être minimisée.

Par conséquent, les protocoles vont interagir les uns avec les autres essentiellement de deux façons différentes [30] :

- ❖ Par le simple échange de paquets de données qui seront transmis d’un composant à l’autre et qui seront traités par les différents protocoles ;
- ❖ Par l’échange d’informations entre les différentes couches, ce qui ne reste pas non plus sans danger [28] car il peut conduire à des boucles de rétroaction ayant des conséquences indésirables sur la performance globale du système.

Nous pouvons donc constater qu’il y aura toujours ce besoin d’optimiser un système, qui risque d’en générer un autre: le compromis entre l’architecture du réseau et ses performances.

Les WSNs restent un cas particulier de réseaux, parce que leur fiabilité implique non seulement la simple livraison de paquets de données (la fiabilité du routage), mais aussi leur capacité de détecter les informations à transmettre (des paramètres physiques par exemple). Pour accomplir ces tâches, il existe un fort besoin de coopération entre les différentes couches (physique, réseau, etc.) ainsi qu’une bonne couverture de la zone surveillée.

Donc, l’intérêt d’adopter une stratégie de conception cross-layer de l’architecture de communication pour les WSNs est presque évident, sans oublier que cette conception doit se réaliser avec beaucoup d’attention.

I.3.2 Les principaux paramètres de performances

Dans la première partie de ce premier chapitre, nous avons débuté avec une présentation générale de WSN, en s’orientant plus particulièrement vers leurs problèmes et par conséquent, vers des méthodes d’optimisation existantes. Nous nous concentrerons maintenant sur l’analyse des performances demandées dans de tels systèmes de communication, qui vont guider nos travaux de recherche vers la conception et l’optimisation des architectures de réseaux de capteurs sans fil.

I.3.2.1 La gestion du réseau

Dans un réseau relativement dense, de nombreux problèmes typiques pour les réseaux sans fil sont accentués par le grand nombre de nœuds voisins :

- ❖ Les nœuds interfèrent les uns avec les autres ;
- ❖ Ils peuvent utiliser inutilement la puissance de transmission pour parler à grande distance et peuvent également limiter la réutilisation de la bande passante ;
- ❖ Il existe beaucoup d'itinéraires possibles et les protocoles de routage peuvent avoir des multiples calculs de routes, même si les mouvements de nœuds correspondent à de petites arrivées ou départs.

Pour surmonter ces nouveaux problèmes, *le contrôle de topologie du réseau* peut être la solution. L'idée est de limiter délibérément l'ensemble des nœuds qui sont considérés comme des voisins pour un nœud quelconque. Cela peut se faire par le contrôle de la puissance de transmission, l'introduction d'une organisation hiérarchique du réseau, la désignation de nœuds avec prise en charge de certaines tâches de coordination, ou tout simplement la désactivation de certains nœuds pendant une période de temps.

Il y a trois grandes possibilités pour faire du contrôle de topologie [64] :

- ❖ Les réseaux basés sur le contrôle de la puissance (Figure I.14 (a)) ;
- ❖ Les réseaux hiérarchiques de type squelette ou backbone (Figure I.14 (b)) ;
- ❖ Les réseaux hiérarchiques divisés en groupes ou clusters (Figure I.14 (c)).

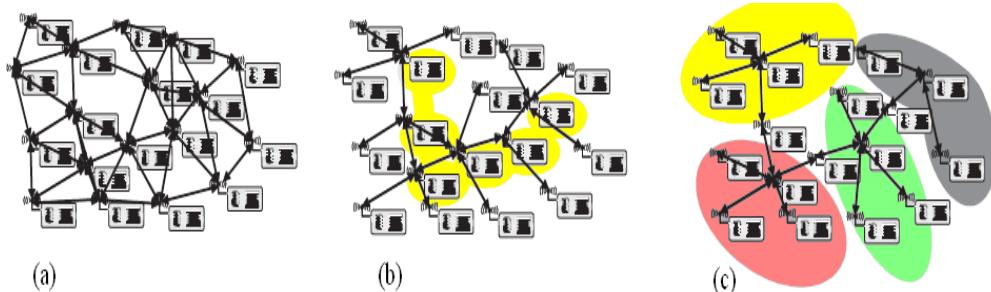


Figure I.14 Différentes options de contrôle de topologie [64].

Il existe également des approches qui combinent ces trois mécanismes et certaines représentent des solutions plus efficaces pour optimiser la topologie d'un réseau sans fil [46], [68], [70].

Les principaux paramètres pour juger l'efficacité d'un contrôle de la topologie, sont : la connectivité, les facteurs d'extension de la consommation d'énergie, la robustesse à la mobilité, ainsi que le coût général d'un algorithme en termes de messages et de calculs [67].

Une autre solution est de faire de la gestion du réseau à travers de différents protocoles de communication. Pour optimiser le fonctionnement d'un réseau de capteurs sans fil ([4] et [63]), des connaissances techniques dans différents domaines sont nécessaires [64]. Plusieurs technologies, comme le Bluetooth, l'IEEE 802.15.4 [64] ou l'UWB [65] ont été utilisées pour mettre en œuvre des réseaux de capteurs. L'UWB semble compétitive dans ce domaine et pourrait être exploitée comme une technologie de transmission prometteuse et flexible.

Comme discuté précédemment, le protocole MAC a un effet significatif sur les performances et la consommation d'énergie dans les réseaux de capteurs. Choisir un support d'accès de type TDMA permet d'éliminer les risques de collisions en raison de l'attribution de time-slots. De plus, en combinant les avantages de TDMA et ceux d'une

topologie divisée en groupes, cela va conduire à une augmentation de l'efficacité dans la consommation d'énergie [66].

Pour des réseaux multi-sauts, il faut également déterminer quelle est la meilleure stratégie de routage. Cela impose au protocole de routage d'être distribué et auto-configurable, pour être en mesure de faire face à des changements fréquents de la topologie de réseau, etc. [64].

En conclusion, le contrôle de la topologie reste un moyen puissant pour l'optimisation du fonctionnement d'un système de communication sans fil, en rajoutant de nouvelles fonctionnalités (un routage sur des graphes différents, des changements dans les relations de voisinage, etc.). En contre partie, son utilisation non judicieuse peut considérablement dégrader les aspects opérationnels, ce qui impose une analyse approfondie des conditions de fonctionnement propres à chaque système. Le choix de la topologie représentera par la suite un aspect essentiel pour l'utilisation de technologies et de protocoles de communications les mieux adaptés dans chaque contexte d'application.

Dans la section suivante nous traiterons un autre aspect concernant les performances d'un système, en particulier la synchronisation dans un réseau de capteurs sans fil.

I.3.2.2 La synchronisation

Depuis que les réseaux de capteurs utilisent un support sans fil, ils héritent tous des problèmes bien connus de la transmission sans fil. Ainsi, selon les schémas de modulation, les fréquences utilisées, la distance entre l'émetteur et le récepteur, et le milieu de propagation, des taux d'erreur binaire (ou le BER (Bit Error Rate)) de l'ordre de 10^{-3} - 10^{-2} peuvent facilement être observés [31] (10^{-2} signifie que l'on a une erreur binaire pour cent bits transmis).

Dans certains cas d'applications, l'envoi occasionnel d'un paquet de données peut être plus que suffisant; dans d'autres cas, il y a des besoins de fiabilité très élevés; et dans d'autres cas encore, la prise en compte du retard est très importante lorsque des actionneurs doivent être contrôlés en temps réel par le réseau de capteurs. Par conséquent, le taux de livraison de paquets est une métrique insuffisante. Il existe donc aussi un besoin de synchronisation dans la transmission des données pour les WSNs.

Selon les exigences des applications, assurer une fonctionnalité de synchronisation de qualité est un défi. Plus précisément, les capteurs doivent recueillir des informations qu'ils transmettent ensuite au collecteur (ou à l'unité centrale). Le collecteur doit recevoir les informations de tous les capteurs dans un délai minimal: c'est le problème de synchronisation décrit dans [48]. Ceci a conduit, dans les réseaux informatiques traditionnels, à la conception de protocoles pour maintenir la synchronisation des horloges physiques. Un protocole tel que NTP (Network Time Protocol) [49] n'est pas un bon choix pour les WSNs en raison d'hypothèses non valides dans un WSN [50], car il réalise la synchronisation d'horloge du serveur central avec l'UCT (Universal Coordinated Time). De plus, même si la synchronisation est de haute précision, elle se fait au prix de la complexité des messages échangés [50].

Plus récemment, la norme IEEE-1588 [52] est devenue la nouvelle référence pour la synchronisation des horloges dans les applications industrielles, de par ses performances de

dizaines de nanoseconde (10ns – 75ns). Mais, comme les réseaux sans fil sont limités en termes de puissance et présentent une grande complexité, la plupart des implémentations de l'IEEE-1588 utilisées sur les réseaux câblés ne s'adaptent pas aux WSNs.

Par conséquent, des travaux pour adapter le fonctionnement d'IEEE-1588 pour les réseaux de capteurs sans fil ont été réalisés. Ainsi, [53] présente des expérimentations et une évaluation de performances pour un protocole de synchronisation basé sur le standard IEEE-1588 pour des réseaux de capteurs sans fil. Les résultats montrent que la synchronisation entre l'horloge maître et les horloges esclaves des nœuds du réseau se réalise avec une précision de quelques centaines de nanosecondes (200ns).

Plusieurs protocoles de synchronisation d'horloge ont été proposés pour les réseaux de capteurs sans fil, avec des performances plus ou moins bonnes. Le RBS (Reference Broadcast Synchronization) [54] est le protocole le plus représentatif. En exploitant la propriété de diffusion de la communication sans fil, ce protocole est capable de réaliser la synchronisation d'un groupe de nœuds qui se trouvent à portée de communication d'un émetteur de référence. Les nœuds qui reçoivent le message de synchronisation, enregistrent l'heure de son arrivée et échangent cette information avec les autres. La précision pour la synchronisation de RBS, en présence de la technologie 802.11 et avec un estampillage du temps réalisé au niveau du noyau du système d'exploitation, est de l'ordre de la dizaine de microseconde [54].

Le protocole TPSN (Timing-sync Protocol for Sensor Networks) est une implémentation de la méthode de synchronisation « émetteur-récepteur ». Dans [55], les auteurs implémentent le principe de TPSN sur une architecture de capteurs de type Berkeley Motes et proposent une procédure d'horodatage au niveau de la couche MAC. Cette méthode est capable de réduire efficacement le temps moyen d'accès au medium (l'erreur moyenne pour la précision de la synchronisation est de 16,9 μ s).

Le protocole FTSP (Flooding Time Synchronization Protocol) proposé dans [56] est un autre protocole de synchronisation qu'utilise la stratégie traditionnelle de synchronisation « émetteur-récepteur ». Ce protocole est similaire à TPSN, dans la mesure où il réalise également une synchronisation de tous les nœuds du réseau à travers un nœud « root », mais en améliorant un certain nombre des défauts. Le nœud « root » est responsable de fournir le temps global dans le réseau et il est régulièrement élu d'une manière dynamique. La topologie du réseau est de type mesh, contrairement à TPSN où l'organisation du réseau est de type hiérarchique.

Dans [58], les auteurs ont évalué le protocole FTSP dans un réseau de 64 éléments organisé dans une grille de 8x8, ce qui correspond à un réseau avec 7 sauts. Après environ 10 minutes, la synchronisation du réseau est réalisée avec une précision moyenne inférieure à 11.7 μ s, d'où une erreur moyenne de 1.7 μ s par saut si nous divisons par le nombre de sauts.

Enfin, le protocole PBS (Pairwise Broadcast Synchronization) proposé dans [57] décrit une nouvelle approche de synchronisation, appelé Receiver-Only Synchronization (ROS). La précision est similaire à celle obtenue pour le RBS (29,1 μ s) sur la technologie Berkeley Motes. PBS nécessite d'un nombre réduit de messages pour chaque cycle de synchronisation, ce qui est un avantage en terme d'économie d'énergie.

1.3.2.3 *La synchronisation vs l'économie d'énergie*

L'objectif concernant la précision de la synchronisation dans les réseaux de capteurs se rajoute donc à d'autres exigences comme l'économie d'énergie, ce qui entraîne une question cruciale: *quels seront le temps et l'énergie nécessaires pour atteindre une bonne précision de la synchronisation ?* De ce fait, les solutions existantes ne sont plus adaptées pour ce type de réseau, ce qui conduit à trouver des méthodes combinées aux autres, afin de pouvoir garantir les services demandés pour chaque application en utilisant un minimum d'énergie. A ce sujet, dans [51] les auteurs proposent un protocole de synchronisation d'horloge qui est économique en énergie, basé sur l'estimation de l'horloge décalée par rapport à une horloge virtuelle.

Globalement, les méthodes traditionnelles sont basées sur une synchronisation des horloges ; plus précisément, elles peuvent fournir à chaque instant un estampillage temporel très exact pour chaque paquet. Mais, dans les cas où le réseau fonctionne en utilisant plusieurs états pour les capteurs (actif, sommeil, etc.), et où l'interface radio est la seule à détenir une source externe de temps, ces méthodes ne sont plus adaptées. Dans ce contexte, les nœuds qui seront en état de sommeil ne vont plus pouvoir se synchroniser et par conséquent le réseau ne pourra pas assurer une bonne précision de la synchronisation.

Cependant, dans [43], est proposée une solution appelée « post-facto Synchronization » qui se préoccupe, en plus de la synchronisation, de la minimisation de l'énergie consommée ; le principe est basé sur l'idée qu'un message nommé « beacon » est envoyé en broadcast à tous les éléments du réseau, ce paquet étant utilisé par la suite comme une base de temps qui servira pour le recalage ultérieur des nœuds capteurs. Dans la même idée, Tian et al font une proposition de synchronisation cette fois-ci au niveau des couches physique et réseau : ils proposent d'intégrer une horloge idéale au niveau de la couche réseau [44]. Le travail se base sur la création d'un nouveau protocole de synchronisation au niveau réseau, basé sur la méthode de Time-Stamp Broadcast Synchronization (TSBS), qui est capable d'effectuer ses tâches avec moins d'énergie consommée.

Les avancées à propos de ce sujet contradictoire (la synchronisation versus l'économie d'énergie) sont assez récentes [59]. Dans [60] et [61], les auteurs proposent un nouveau système de synchronisation des horloges afin d'optimiser la consommation énergétique et comparent les résultats avec celle du protocole TPSN. Pour répondre à la nécessité des nœuds mobiles de se synchroniser dans un réseau, [62] propose une solution hybride qui intègre les protocoles RBS et TPSN.

Nous avons observé à ce sujet que le protocole PBS est le seul protocole de synchronisation qui, en raison de son « modus operandi », est capable de réaliser des économies d'énergie importantes, hormis ses performances de précision des horloges.

La satisfaction de ces deux besoins, la gestion d'énergie et la synchronisation, n'est pas une tâche facile étant donné que les critères de performances sont opposés. Plus précisément, pour assurer une bonne synchronisation, le réseau va consommer une quantité importante d'énergie. Par conséquent, il va falloir trouver le compromis qui sera capable de garantir les performances ainsi qu'une longévité du réseau.

1.3.2.4 L'économie d'énergie

Une grande partie de la discussion précédente montre que l'énergie est une ressource précieuse dans les réseaux de capteurs sans fil et que l'efficacité énergétique doit être un objectif évident d'optimisation [45].

Dans de nombreux scénarios, les nœuds capteurs devront compter sur un approvisionnement limité en énergie (la batterie). Le remplacement de ces sources d'énergie sur le terrain n'est généralement pas facile, et dans le même temps, un réseau de capteurs sans fil doit fonctionner aussi longtemps que possible. Par conséquent, la durée de vie d'un WSN devient un aspect crucial.

Comme alternative à l'approvisionnement énergétique, une source d'alimentation via des cellules solaires, par exemple, pourrait être envisagée sur un nœud capteur. Typiquement, ces sources ne sont pas assez puissantes pour assurer un fonctionnement continu, mais peuvent fournir une certaine recharge des batteries et dans ces conditions, la durée de vie du réseau pourrait être améliorée. Mais en fonction du contexte, cette alternative n'est pas physiquement réalisable (le manque de luminosité par exemple), ce qui impose de trouver d'autres solutions, comme des stratégies basées sur le calcul informatique ou une conception de type cross-layer.

Une autre méthode permettant un gain considérable en consommation d'énergie est d'éteindre l'interface de communication quand celle-ci n'est pas utilisée. Mais, il ne faut pas oublier que la taille des paquets échangés dans un réseau de capteurs est petite. Dans ces conditions, si l'interface de communication est éteinte pour chaque période de passivité, la quantité d'énergie consommée va être plus importante que si l'interface était laissé allumée en continu [37].

Un autre aspect à ne pas oublier est la puissance de rayonnement de l'émetteur-récepteur. L'augmentation de cette puissance augmente l'énergie consommée par bit, donc le SNR (Signal to Noise Ratio), et diminue ainsi le taux d'erreur binaire et le besoin de retransmissions. Ebert et Wolisz [38] montrent que pour un réseau ad hoc avec un seul saut, nous avons une puissance d'émission plus optimale si nous utilisons des mécanismes d'harmonisation de l'énergie rayonnée, pour une longueur de trame donnée. Dans [46], une solution de contrôle pour la transmission d'énergie pour les nœuds capteurs est proposée afin de prolonger la durée de vie du réseau, sans affecter la fonctionnalité du système.

Par contre, dans un grand réseau de communications multi-sauts, les choses vont être différentes. Si un nœud augmente sa puissance d'émission, il augmente aussi le risque des interférences avec ses voisins et donc aussi le taux d'erreur binaire qu'ils ont à traiter. Ce phénomène s'appelle surcoût ou « overheard », qui conduit au final à une importante perte d'énergie.

Dans le but d'assurer différentes fonctionnalités comme la connectivité ou l'évitement de collisions, la plupart des protocoles MAC intègrent des boucles de contrôle, qui envoient périodiquement des messages spécifiques. Cette surcharge représente une source de consommation d'énergie additionnelle qui réduit la vie du système. Il n'y a même pas un compromis à faire, car renoncer à de telles fonctionnalités peut conduire à la fin de vie du réseau. La meilleure stratégie dans cette situation serait d'intégrer ce trafic de surcharge dans le trafic utile.

Un autre phénomène qui peut apparaître au cours d'une transmission des données est la surémission (overemitting). Ce phénomène se produit quand le destinataire ne peut pas recevoir et donc les messages envoyés deviennent inutiles énergivores [47].

Si nous faisons un rappel des principales sources impliquées dans la consommation énergétique d'un capteur, nous retrouvons : en premier, la capture des informations, puis le traitement et enfin la transmission des données. Les études montrent l'intérêt pour l'aspect énergétique, comme [40], dans laquelle les auteurs proposent un modèle de la consommation énergétique, qui prend en compte les énergies pour émettre et recevoir, l'énergie de transmission électrique et d'amplification.

Narendran et al. [39] décrivent un système entièrement distribué dans lequel la puissance de l'émetteur s'adapte à partir des mesures des taux d'erreur et des interférences, et où la puissance reçue au niveau du récepteur est renvoyée à l'émetteur. L'algorithme est développé et évalué dans un scénario de système cellulaire, où toutes ces informations sont facilement disponibles dans les stations de base. Toutefois, l'approche est suffisamment générique pour être appliquée aussi dans les réseaux de capteurs sans fil.

D'autres méthodes adoptées pour la gestion d'énergie consiste à optimiser le fonctionnement des protocoles de niveau MAC.

Ainsi, dans [41], les auteurs s'intéressent à modéliser un protocole MAC UWB qui assure le contrôle de la puissance d'émission et l'ordonnancement (ou « scheduling »). L'optimisation globale du réseau est ainsi réalisée par un protocole de routage multi-sauts. Ils montrent aussi que la réalisation d'un protocole MAC optimal est indépendante du protocole de routage utilisé.

Basés sur des résultats théoriques, Merz et al [42] proposent un protocole MAC pour la technologie UWB dans laquelle, en mode constant, « la source » fait une adaptation pour le codage du canal avec le niveau d'interférence de la destination. La stratégie consiste à faire le choix du taux de transmission le mieux adapté. Pour diminuer les interférences, ils proposent un scénario de démodulation spécifique qui consiste à éteindre les sources de communication qui interfèrent.

Dans tous les cas, comme le nombre d'éléments dans un seul réseau de capteurs est très grand, la notion de redondance spatiale est très importante pour pouvoir assurer une grande couverture et maximiser la durée de vie du réseau. Par conséquent, cela demande une bonne gestion de l'énergie des capteurs depuis leur conception jusqu'à leur déploiement.

I.3.2.5 Topologie dynamique et passage à l'échelle

Dans un WSN, où les nœuds sont généralement déployés de manière redondante pour pouvoir se protéger contre les possibles défaillances ou pour compenser une « mauvaise qualité » offerte par un seul équipement de détection, l'identité propre d'un nœud capteur devient sans importance. Ce qui est important, ce sont les données en elles-mêmes. Par conséquent, *le passage d'un paradigme centré sur l'adresse à un autre basé sur les données* est plus pertinent dans la conception de protocoles pour WSNs.

Dans un WSN, le nombre de nœuds par unité de surface (*densité du réseau*) peut varier considérablement. Et au sein de la même application, la densité peut varier dans le temps et dans l'espace, car les nœuds deviennent défaillants ou se déplacent. Par

conséquent, le réseau doit s'adapter à ces variations. Dans [4] par exemple, la densité moyenne des nœuds capteurs dans un réseau est estimée à environ 20 capteurs/m³.

Finalement, une manière fixe de traiter l'information est insuffisante dans un réseau de capteurs. Il sera donc nécessaire pour les nœuds, non seulement de traiter les informations, mais aussi de réagir avec souplesse sur les changements de leurs tâches. Les nœuds doivent être programmables, et leur *programmation doit être modifiable en cours de fonctionnement* lorsque de nouvelles tâches deviennent importantes.

Sachant que l'environnement et les WSNs eux-même peuvent évoluer en temps (l'épuisement des piles, des nœuds défaillants, de nouvelles tâches à accomplir), le système doit s'adapter. Il doit surveiller son propre fonctionnement et pouvoir modifier les paramètres d'exploitation ou faire différents compromis (par exemple, fournir une qualité inférieure lorsque les ressources énergétiques diminuent). Dans cette situation, *le réseau doit s'autogérer* et également être en mesure d'interagir avec les mécanismes externes de maintenance pour assurer son fonctionnement prolongé avec une qualité satisfaisante pour l'application [86].

Comme la richesse des options d'une interface générale pour les réseaux de capteurs sans fil est vaste, les architectures et les protocoles utilisés doivent être en mesure de *passer à l'échelle* (ou l'agrandissement du réseau) et pouvoir fonctionner dans beaucoup de conditions différentes.

Ainsi, trois éléments clés vont émerger : la concentration des données, les compromis par rapport à l'énergie et la précision de la synchronisation des données, ce qui rend encore une fois ces réseaux différents des autres. Il n'est donc pas surprenant que toutes les problématiques ne soient pas encore totalement résolues.

Ce sont ces problèmes qui vont être abordés dans les chapitres suivants, dans le but d'être améliorés. Mais tous d'abord, nous allons nous intéresser dans la section ci-dessous aux méthodes d'évaluation existantes, qui vont nous permettre de réaliser une meilleure analyse des performances et ainsi une optimisation du système.

I.4 Evaluation des WSNs

Un réseau de capteurs est composé de dispositifs similaires de détection qui sont configurés avec les mêmes composants. La configuration mémoire de tous les capteurs est donc souvent identique. Comme introduites auparavant, les exigences demandées en fonction de l'application peuvent être très différentes. Ainsi, avant de démarrer la phase de réalisation et de déploiement d'un réseau si complexe, des étapes de vérification, de test et de validation s'imposent.

Une approche possible peut être de modéliser le comportement et les caractéristiques des réseaux de capteurs comme un ensemble de représentations de type UML (Unified Modeling Language), qui se traduit en «Sensor Modeling Language» (**SensorML**¹³). Ce langage a été créé pour des applications spécifiques (les géosciences, l'observation de satellites de la terre) ; il se concentre principalement sur la description des capacités des capteurs individuels. En parallèle, il prévoit des méthodes pour exprimer, par exemple, la précision d'acquisition et le traitement des données. Le principal avantage est ici la possibilité de décrire explicitement le «sens» des paramètres mesurés.

¹³ <http://www.opengeospatial.org/standards/sensorml>

Une autre approche est de faire appel à un outil de simulation qui permet de réaliser une analyse complexe et de prendre en compte de manière assez fidèle l'influence des paramètres de réseau, dans différentes conditions.

Pour une analyse pertinente d'un réseau avant qu'il ne soit construit et que tous ses éléments ne soient disponibles, une émulation du réseau pourrait être également une bonne alternative.

Ainsi, dans la section suivante nous détaillerons les environnements de simulation et d'émulation existants les plus utilisés/connus dans la littérature, avec les avantages et leurs inconvénients, dans le but d'identifier l'outil le mieux adapté à nos besoins.

I.4.1 Etude des outils disponibles

Avant de commencer la phase de déploiement effectif d'un WSN, une étape de simulation est nécessaire. Cette étape va apporter une validation du bon fonctionnement du système, et par conséquent une minimisation des erreurs de conception, qui joue un rôle important dans le coût final de construction.

En ce qui concerne les outils de simulation/émulation existants, le choix à faire est souvent difficile, car les possibilités présentes sont multiples. Ainsi, nous devrons tout d'abord définir des critères de choix plus génériques comme : la meilleure adaptabilité pour simuler un WSN, les avantages offerts à l'utilisation, la popularité, les performances en terme des résultats, etc. En même temps, il faut impérativement tenir compte des principales contraintes et besoins propres à l'application envisagée. Dans ce sens, nous annonçons les points forts caractéristiques à notre système : concevoir un réseau de capteurs sans fil avec des centaines d'éléments et garantir une bonne couverture, assurer une très bonne précision de la synchronisation des données, réaliser une meilleure gestion de l'énergie.

SIMULATEURS

Un logiciel de simulation est un programme qui permet à l'utilisateur d'observer une opération grâce à la simulation, sans réellement l'effectuer.

Une simulation permet d'examiner le comportement du système dans différents scénarios. Elle peut être utilisée pour identifier des problèmes de logique dans le processus, pour offrir une plateforme sûre et efficace (en terme de coûts et de temps), pour évaluer les effets secondaires et pour optimiser les performances du système tout ceci avant la réalisation physique du système.

WORLDSSENS [72]

Worldsens est une approche de développement et simulation dédiée à la conception, qui réunit deux outils, **WSim**¹⁴ et **WSNet**¹⁵, qui peuvent être utilisés conjointement ou individuellement. Le besoin d'un tel type d'outil est très important pour la conception de réseaux de capteurs, car ils permettent de dimensionner, programmer et estimer les paramètres de performances pour une application distribuée embarquée.

¹⁴ <http://perso.ens-lyon.fr/eric.fleury/Upload/wsim-docbook/>

¹⁵ <http://wsnet.gforge.inria.fr/index.html>

Pour réaliser à travers l'architecture de Worldsens une simulation conjointe avec ces deux types de composants (matériel et logiciel), il faut faire le choix du matériel qui va être implémenté sur WSim et après choisir la partie logicielle à travers le WSNet.

Vis-à-vis d'autres simulateurs, Worldsens offre des propriétés de flexibilité et de passage à l'échelle, car une simulation peut être distribuée sur différentes machines. Un autre aspect important concerne la synchronisation qui est assurée entre la simulation matérielle de nœuds avec WSim et la simulation de l'interface radio avec WSNet, ce qui permet de résoudre les problèmes liés aux contraintes temps réel.

Le besoin d'étudier le profil d'exécution d'application est résolu aussi par l'option de sauvegarde des traces de la fréquence d'horloge, de l'état interne des périphériques, de la consommation d'énergie, etc. Autrement dit, en utilisant ces logiciels réunis par Worldsens, le concepteur peut réaliser toutes les étapes de développement d'une application, à partir de la conception et du dimensionnement, jusqu'à la phase de validation et d'optimisation.

WSim est un outil de bas niveau qui est dédié à la simulation du matériel, à l'aide de microcontrôleurs. Il assure une simulation complète et fournit une analyse exacte de l'exécution des événements dans la plateforme.

Le point positif dans son fonctionnement est que le temps de référence calculé pour la simulation est de l'ordre de la nanoseconde, ce qui favorise des rapports de fréquence très précis entre les différentes sources. En contrepartie, cette approche ne permet pas pour l'instant de traiter des architectures multiprocesseurs.

Le **WSNet** se charge de la simulation logicielle pour un réseau de capteurs. Il est compatible avec Linux et Windows (à travers Cygwin). Il est constitué à la base d'un fichier de type XML dans lequel le programmeur doit spécifier les conditions de simulation (la durée, le nombre de nœuds, les modules utilisés, les bibliothèques, etc.). Ensuite, il faut donner une description de l'environnement de la simulation, ce qui signifie le type du canal radio. Enfin, il ne faut pas oublier la description de l'architecture des nœuds simulés, ça veut dire les modules utilisés et avec leur interactions.

OPNET¹⁶

Opnet est un outil de simulation largement utilisé dans le secteur commercial. Il permet de concevoir et de simuler des protocoles et des applications pour des réseaux de communication, à travers une interface programmée en C++.

Au fil du temps, il a été répertorié comme peu fiable par les études comparatives qui ont été réalisées [75]. Ainsi, son comportement se montre divergent de point de vue quantitatif et qualitatif.

OMNeT++¹⁷

Dans cette taxonomie des outils de simulation, nous ne pouvons pas oublier une bibliothèque de composants basés C++, extensible et modulaire, nommée OMNeT++. Elle utilise le langage NET (compatible XML) pour définir la structure, la topologie (mesh, étoile, ring, tree, etc.) et les paramètres (les délais de propagation, les débits de données, les taux d'erreur, etc.) de la simulation. À travers l'éditeur graphique NET, cette bibliothèque permet la création de topologies entièrement paramétrables.

¹⁶ <http://www.opnet.com/>

¹⁷ <http://www.omnetpp.org/>

De plus, ce simulateur offre un très bon temps d'exécution. Par contre, le fait qu'il soit conçu par événements discrets et non en temps réel, rend difficile la simulation de plusieurs événements en même temps. Un autre inconvénient est que l'utilisateur n'a pas accès à des informations comme l'architecture interne des nœuds ou l'implémentation des différents modules, l'intérêt restant à un niveau plus élevé (comme par exemple choisir et dimensionner les protocoles de la couche MAC).

NS-2¹⁸

Network Simulator est un simulateur à événements discrets pour les réseaux qui est principalement utilisé pour la simulation des protocoles de tous les niveaux de communication, offrant un support pour les réseaux filaires, aussi bien que sans fil. Il a été construit en C++ et fournit une interface de simulation à travers OTcl, un langage orienté objet de Tcl. L'utilisateur doit décrire la topologie de réseau en écrivant des scripts OTcl, qui sont ensuite exécutés par le programme principal NS-2.

Toutefois, la modélisation dans NS-2 reste une tâche complexe : il n'y a pas d'interface graphique et il faut apprendre des notions comme la théorie de files d'attente, le langage de script et les techniques de modélisation. L'avantage est qu'une fois ces aspects maîtrisés, le caractère open-source de cet environnement de simulation donne une grande liberté à l'utilisateur (optimisation des protocoles déjà implémentés et modélisation des nouveaux, accès et maîtrise de toute la pile protocolaire pour faire du cross-layer, etc.).

NS-3¹⁹

C'est une évolution du simulateur NS-2, dédiée pour les systèmes qui utilisent l'interface de communication de la pile Internet, en s'adressant principalement à la recherche et à l'utilisation pédagogique. Les programmes de simulation sont, cette fois ci, des exécutables C++ ou Python. La principale différence par rapport à la version NS-2 est que NS-3 est conçu pour utiliser le vrai noyau du système d'exploitation avec ses applications (des échos serveurs, des générateurs de trafic).

Il présente l'inconvénient majeur de ne pas pouvoir gérer une structure hiérarchique du réseau, ce qui amène à moins de flexibilité pour l'architecture du modèle simulé.

EMULATEURS

Une alternative à la simulation peut être « l'émulation », qui consiste en un banc d'essai avec des installations matérielles et logicielles. Cela donne aux chercheurs un éventail d'environnements qui leur permet de développer, déboguer et évaluer leurs systèmes.

ATEMU²⁰

ATEMU est un logiciel d'émulation qui peut être directement utilisé par les utilisateurs de système TinyOS : il est compatible avec du matériel Mica2 et peut être facilement étendu à d'autres plates-formes de capteurs. TinyOS²¹ est un système

18 <http://www.isi.edu/nsnam/ns/>

19 <http://www.nsnam.org/doxygen-release/index.html>

20 <http://www.cshcn.umd.edu/research/atemu/>

21 TinyOS Mission Statement. UC Berkeley, 2004. <http://www.tinyos.net/>

d'exploitation open-source qui a été conçu spécialement pour des applications avec des réseaux de capteurs sans fil, car son principe de fonctionnement est basé sur la réduction de la consommation en énergie du capteur (il est programmé pour se mettre en veille lorsqu'il n'y a pas des tâches actives). ATEMU est la combinaison des deux éléments: l'émulateur de base et le débogueur graphique **Xatdb**²². Cette fenêtre graphique d'ATEMU permet de visualiser tous les évènements associés à un capteur choisi, ce qui permet de simuler et concevoir des applications fiables.

Un avantage est qu'il fournit une excellente plate-forme pour effectuer des comparaisons objectives entre nombreux protocoles de communication et que les résultats sont beaucoup plus réalistes que pour d'autres simulateurs. A l'inverse, même s'il permet de simuler un réseau de capteurs d'une manière assez fidèle, des problèmes de mémoire pour un grand nombre d'éléments apparaissent.

AVRORA²³

Un autre émulateur des réseaux de capteurs sans fil, écrit dans le langage Java, est AVRORA. Il est capable de simuler un réseau de plusieurs nœuds, en exécutant des programmes déchargés de TinyOS sur les plateformes de Mica2 or Micaz. AVRORA réalise une simulation complexe du microcontrôleur, des périphériques et de la communication radio.

En ce qui concerne les inconvénients, nous pouvons noter que tous les nœuds sont simulés dans un seul processus d'exécution et donc sur un même ordinateur, cette approche ne permettant pas un passage à l'échelle efficace. Une autre limitation importante est qu'il n'implémente pas certains composants principaux comme un gestionnaire d'horloges, dû à l'incapacité de modéliser une dérive d'horloge entre les nœuds ou entre les composants internes d'un même nœud.

TOSSIM [71]

TOSSIM est un autre émulateur qui fonctionne en présence du système d'exploitation TinyOS Networks, mais qui reste compatible aussi avec d'autres solutions, comme Linux, Windows ou Mac OS. TOSSIM est capable de simuler le code spécifique de l'utilisateur qui tournera sur une plateforme réelle, et permet ainsi de tester et valider le comportement des éléments dans le même réseau. Les principales étapes pour créer une application en TOSSIM sont: la description d'un chemin en XML, la compilation de l'application TinyOS, la compilation de l'interface programmée et la construction des objets et du support Python. Tous ces éléments donnent une idée sur la complexité et la difficulté de prise en main de cet émulateur.

La présence de l'interface TinyViz²⁴ offre la possibilité de ralentir la simulation par un délai, afin d'observer le déroulement des évènements, ce qui est très intéressant lorsque le réseau est surchargé de messages. Cependant, il a l'inconvénient majeur que ce simulateur ne permet pas une capture des informations liées aux interruptions ou à la temporisation dans le code, ce qui met des limites au niveau l'analyse de la précision pour la synchronisation des données.

²² <http://www.hynet.umd.edu/research/atemu/userdoc/index.html>

²³ <http://compilers.cs.ucla.edu/avrora>

²⁴ <http://www.tinyos.net/tinyos-1.x/doc/tutorial/lesson5.html>

Il faut rajouter qu'il existe aussi une extension, le **PowerTOSSIM** [72], qui intègre un modèle d'énergie, ce qui apporte des informations sur la consommation des nœuds, en fonction de leur état. Cependant, cette version n'est compatible qu'avec des applications qui utilisent le système d'exploitation TinyOS.

PLATEFORMES D'EMULATION

Une plate-forme classique est **Emulab**²⁵. Son installation principale est gérée par l'Université de l'Utah, sans oublier les douzaines de sites dans le monde entier, qui réunissent des bancs d'essais avec des centaines de nœuds. Emulab est principalement utilisé par les chercheurs en sciences informatiques dans les domaines des réseaux et des systèmes distribués. Il est également conçu pour émuler des petits réseaux des capteurs, car il est équipé avec de capteurs Mica2.

SensLab²⁶ est une autre plateforme de tests pour des applications réelles, qui réunit plusieurs sites en France : Lille, Strasbourg, Grenoble et Rennes, avec un total de 1020 nœuds, environ 250 nœuds pour chaque site. Cette plateforme est caractérisée par une grande flexibilité (il n'y a pas de contraintes au niveau de la programmation des nœuds, pas d'OS imposé, accès en temps réel, la possibilité de lancer plusieurs applications simultanément, etc.) et une accessibilité fiable à distance.

Les efforts de choisir la meilleure solution pour tester et valider le fonctionnement d'un réseau sont motivés par l'idée que cette étape reste obligatoire pour la réalisation pratique d'une architecture optimale de communication. Un logiciel de simulation/émulation reste largement utilisé pour la conception des équipements, de telle façon que le produit final sera aussi proche de la conception réelle, sans impliquer des modifications coûteuses.

Donc, un des aspects importants pour notre étude est de choisir l'outil le mieux adapté pour un réseau de capteurs sans fil. Comme il existe de nombreuses solutions basées sur des caractéristiques et des objectifs différents, nous présenterons dans le tableau suivant une synthèse de notre étude :

| Simulateurs | Caractéristiques |
|----------------------------------|--|
| OPNET (1986) | +Environnement interactif permettant la conception et l'étude des réseaux ; -Peu fiable de point de vue qualitatif et quantitatif. |
| ATEMU 0.4 (2004) | +Excellente plate-forme de comparaisons, compatible MICA2 et autres ; débogueur graphique Xatdb. -Des problèmes pour un grand nombre d'éléments dans le réseau. |
| Worldsens (2007) | +Outil de prototypage virtuel, offre des propriétés de flexibilité, de passage à l'échelle et de sauvegarde de traces avec la fréquence d'horloge. -Très complexe. |
| AVRORA Beta 1.7.106 (2008) | +Simule avec succès des réseaux de plusieurs nœuds ; compatible TinyOS sur les plateformes AVR. -Ne permet pas un passage à l'échelle efficace, incapacité de modéliser une dérive d'horloge. |

²⁵ <http://www.emulab.net/>

²⁶ <http://www.senslab.info/>

| | |
|------------------------|---|
| OMNeT++ 0.4 (2009) | <ul style="list-style-type: none"> +Simule des topologies paramétrables avec un très bon temps d'exécution. -Incapable de simuler plusieurs événements en même temps, pas d'accès à des informations comme l'architecture interne des nœuds ou l'implémentation des différents modules. |
| NS-2 2.34 (2009) | <ul style="list-style-type: none"> +Simulateur open-source à événements discrets pour tous les niveaux de communication, donne une grande liberté à l'utilisateur. -Complexe. |
| TOSSIM 2.1.1 (2010) | <ul style="list-style-type: none"> +Logiciel open-source, spécialement conçu pour les WSNs, basé sur le besoin d'économie d'énergie. L'interface TinyViz ralentit la simulation par un délai. -Limité dans l'analyse de la précision pour la synchronisation des données. |
| PowerTOSSIM (2010) | <ul style="list-style-type: none"> +Dispose d'un modèle d'énergie pour les nœuds en fonction de leur état. -Compatible qu'avec des applications qui utilisent l'OS TinyOS. |
| NS-3 3.10 (2011) | <ul style="list-style-type: none"> +Dédié aux systèmes de communication Internet, utilise le vrai noyau du système d'exploitation. -Ne gère pas une structure hiérarchique, ce qui donne moins de flexibilité pour l'architecture du modèle simulé. |

Tableau I.2 Comparaison des différents outils de simulation.

Il existe également d'autres simulateurs/émulateurs qui ne rentrent pas dans notre classification, par ce qu'ils ne sont pas adaptés au contexte de réseaux de capteurs sans fil.

Parmi tous les outils étudiés, nous nous sommes arrêtés sur les concepts de simulation proposés par le logiciel NS-2. Ce logiciel est reconnu et particulièrement utilisé dans l'étude de la nature dynamique des réseaux de capteurs sans fil.

I.4.2 Justification de notre choix

Le simulateur NS-2 est un logiciel disponible dans le domaine public avec une utilisation gratuite, exécutable également sous Linux et Windows. À travers ce logiciel, nous avons la possibilité de tester une multitude de protocoles de différents niveaux et dans des réseaux différents (local, filaires, sans fil, satellitaire, etc.). Le simulateur possède une palette de systèmes de transmission directement exploitable, mais donne également la possibilité aux utilisateurs d'implémenter et tester leurs propres systèmes et protocoles. Les principaux composants actuellement disponibles sont présentés dans le Tableau I.3:

| Niveau de communication | Composants |
|---------------------------|---|
| Application | Générateurs de trafic, web, ftp, Telnet |
| Transport | TCP, UDP, RTP, SRM |
| Routage | Statique, dynamique, multipoint |
| Gestion de file d'attente | DropTail, RED, Token bucket ²⁷ |
| Discipline de service | CBQ, SEQ, DRR, Fair queueing |
| Système de transmission | CSMA/CD, CSMA/CA, Point à point |

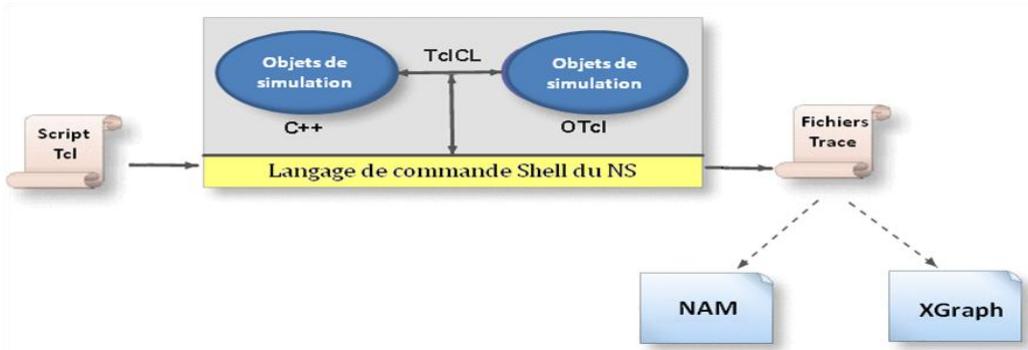
²⁷ Algorithme utilisé pour contrôler la quantité de données qui est « injectée » dans un réseau

Tableau I.3 Les principaux composants du simulateur NS-2.

NS-2 se base sur deux langages de programmations distinctes (Figure I.15):

- ❖ C++ qui constitue la partie centrale du simulateur (le noyau) et qui définit tout le mécanisme interne des objets de simulation;
- ❖ OTcl (Object-oriented Tool Command Language) qui met en place la simulation par l'assemblage et la configuration des objets ainsi que la planification des événements discrets.

La liaison entre les deux langages se réalise à travers des scripts TclCL (Terminal Control Language with classes - l'équivalent d'une interface applicative d'un programme écrit en C++). Tcl est un langage de commande du même type que le Shell UNIX mais qui a été conçu pour contrôler les applications ; autrement dit, l'utilisateur peut définir ici son scénario de simulation. Pour une visualisation et une analyse de la simulation, nous avons à la disposition un outil qui s'appelle NAM²⁸ (Network Animator). Il peut afficher une image avec la topologie du réseau, la dynamique du déroulement du script Tcl dans le temps, etc. Le simulateur permet aussi l'exploitation des fichiers des données par un traitement réalisé avec le traceur de données générales XGraph.

**Figure I.15 L'architecture de base pour le simulateur NS-2.**

Nous pouvons donc constater que ce simulateur offre une ouverture pour l'étude de nombreux mécanismes au niveau des différentes couches de communication. En général, NS-2 offre à ses utilisateurs un moyen d'utiliser les protocoles déjà existants et simuler leurs comportements correspondants, ainsi que rajouter de nouvelles fonctionnalités.

En temps qu'utilisateur, nous constatons que la maîtrise du simulateur à partir du langage Tcl est plutôt facile mais il devient plus compliqué de comprendre et modifier le noyau écrit en C++. Comme NS-2 offre un grand nombre d'objets C++, il est conseillé d'utiliser ces objets et de créer une simulation en utilisant un script de simulation Tcl. En fonction de l'application, nous trouvons l'apport de ces objets insuffisant et donc ça implique un travail supplémentaire pour développer des objets spécifiques et d'utiliser une interface de configuration OTcl pour les rendre fonctionnels.

De plus, dans notre étude les couches de communication concernées sont multiples (Physique, MAC, Transport, Réseau, Application et même le Canal de transmission) dans le but de traiter toutes les contraintes du réseau et d'offrir par la suite un fonctionnement optimal pour le système envisagé. Cela a conforté notre choix vers l'outil de simulation NS-2.

I.5 Objectifs de notre recherche

²⁸ <http://www.isi.edu/nsnam/nam/>

Cette section a comme but de présenter les principes fondamentaux de la transformation des nœuds capteurs individuels vers un réseau de capteurs sans fil. Dans ce contexte, nous présentons tout d'abord notre application de capteurs sans fil et par la suite les objectifs d'optimisation résultant de la façon dont un tel réseau devrait fonctionner.

Ainsi, nos principaux objectifs seront : de concevoir un système autonome multi capteurs robuste et avec un routage efficace, d'assurer une bonne précision pour la synchronisation des données, ainsi que de garantir une efficacité pour la consommation d'énergie.

À la fin de ce chapitre, le lecteur aura à disposition une vision générale de la conception et de la mise en œuvre de réseaux de capteurs sans fil.

I.5.1 Présentation générale de notre application

Nos travaux concernent donc des classes d'application qui permettent :

- ❖ De surveiller en temps réel un système embarqué critique ;
- ❖ D'acquérir des données physiques (température, pression, vibration, évènement, etc.) ;
- ❖ De faire un traitement local plus ou moins complexe ;
- ❖ De mémoriser ou stocker les données pendant un certain temps et de les transmettre de façon sécurisée à un système maître (Ordinateur, PC Portable, PDA etc.) afin de donner des informations fonctionnelles sur le système.

Accomplir toutes ces tâches impose le choix de technologies innovantes (microélectronique, informatique, etc.) et demande beaucoup de compétences.

Les applications exécutées par ce type de système sont : des mesures en phases d'initialisation et surveillance de fonctionnement, de synchronisation, de détection des anomalies et prévention des pannes, de reconfiguration à la demande.

Parmi les principaux domaines envisagés par cette application, nous trouvons l'aéronautique et le spatial, mais aussi le domaine automobile ou ferroviaire. Pour préciser nos objectifs et les conforter par des éléments pratiques, nous avons souhaité utiliser un exemple issu d'un de ces domaines.

L'exemple que nous avons choisi est issu des informations recueillies dans le cadre contractuel du projet aéronautique SACER²⁹ qui a pour but de concevoir un système autonome de capteurs sans fil. En aéronautique, ce réseau de capteurs embarqué sans fil est utilisé pour le suivi en temps réel des grandeurs environnementales telles que la température et la pression, ceci via les capteurs disposés à plusieurs endroits de ailes d'avion. Une telle topologie simplifiée est présentée sur la Figure I.16 :

²⁹ <http://www.intespace.fr/page/l-entreprise/1013-nos-partenaires.html>.

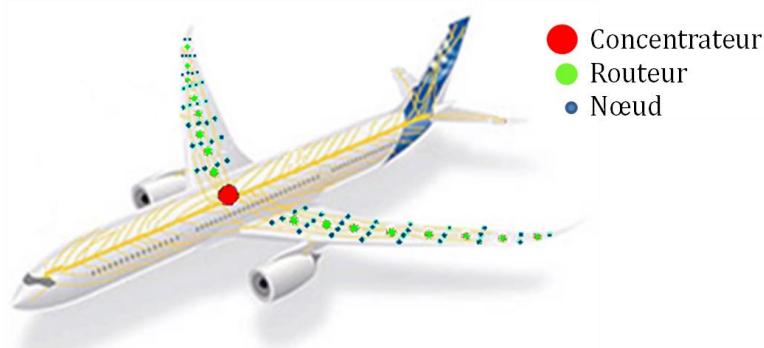


Figure I.16 Réseau de capteurs embarqués sans fil.

Pour la mise en réseau de notre application, en plus des nœuds capteurs qui sont les éléments de base d'un système embarqué, d'autres éléments sont généralement nécessaires pour l'implémentation finale du réseau, comme le concentrateur et les routeurs que nous allons décrire succinctement dans la partie suivante.

I.5.2 Le choix de la topologie

Notre système est un réseau composé d'un concentrateur communiquant avec 8 routeurs, qui communiquent individuellement avec 8 nœuds, comme le montre la Figure I.17. Cette organisation est identique pour chaque aile d'avion (Figure I.16), ce qui signifie un nombre total de 147 éléments communicants (1 concentrateur + 8 routeurs + 8*8 nœuds par aile). Comme le choix en terme de nombre des routeurs n'a pas été imposé par l'application, nous avons fait une analyse des plusieurs scénario de configuration qui nous a conduit à la solution optimale de 8 routeurs par aile.

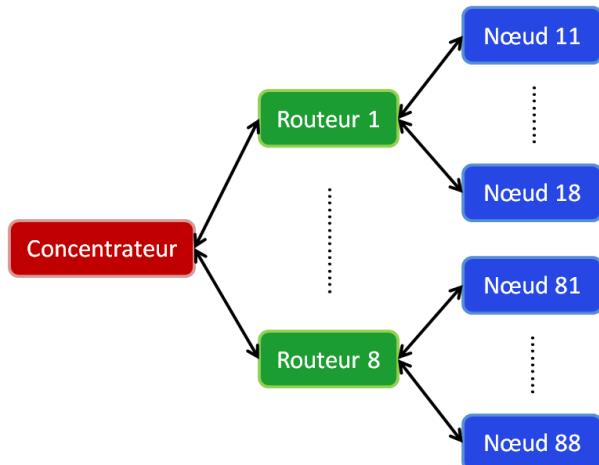


Figure I.17 Topologie du réseau.

Dans le mode de fonctionnement du réseau, la principale caractéristique est que la communication est centrée sur les données, ce qui signifie que des requêtes humaines/opérateurs sont diffusées dans tout le réseau. À leur tour, les nœuds capteurs commencent à recueillir des informations concernant la demande ; puis les informations font le chemin inverse et les routeurs agrègent les données provenant de capteurs et les envoient vers le concentrateur.

Ainsi, la notion de diffusion directe va impliquer des requêtes qui spécifient les besoins de l'utilisateur ou du créateur de requêtes.

Le choix de cette organisation hiérarchique, comme nous avons déjà vu, permet d'économiser l'énergie. Par conséquent, cette stratégie porte son avantage dès le début sur la conservation d'énergie totale du système.

Une fois la topologie du système établie, nous allons maintenant annoncer les principaux problèmes qui constituent le centre d'intérêts de nos travaux de recherches.

Une des principales contraintes dans notre application est le besoin de prise de mesures quasi-synchrones. Plus précisément, le décalage entre deux mesures sur des capteurs différents doit être situé entre $1\mu\text{s}$ et 500ns , d'où en découlent des contraintes d'estampillage comparables.

Comme les nœuds capteurs et éventuellement les routeurs vont être alimentés par batteries, un autre aspect très important à traiter est la gestion de l'énergie totale du système. Avec le choix d'une topologie hiérarchique nous pensons réaliser des économies d'énergie dès le départ, mais ce n'est pas une solution sur laquelle nous nous reposons en ce qui concerne cet aspect. Pour cela, nous avons étudié plusieurs stratégies que nous présenterons dans les sections suivantes.

Enfin, imaginer le routage le mieux adapté à ce contexte applicatif va nous permettre d'avoir un système fiable, capable de répondre à tous les besoins de futurs utilisateurs.

I.5.3 Le besoin de synchronisation

Comme nous avons pu le constater, les WSNs et en particulier les réseaux dédiés à l'instrumentation et à la mesure, ont un certain nombre de caractéristiques qui les différencient des autres types de réseaux. A partir de ce raisonnement, l'élément indispensable pour la détection, *le capteur*, présente des propriétés spécifiques, mais doit de plus faire face à l'interaction avec l'espace et le temps, ce qui se traduit par l'emplacement, la couverture et la synchronisation des données.

Les modules responsables de l'aspect synchronisation d'horloge sont les logiciels pilotes de communication de la couche physique. Ils doivent gérer aussi des liaisons de transmission radio à canaux multiples, y compris l'encodage/décodage du signal et la modulation.

Il existe de nombreux protocoles qui ont été conçus pour résoudre le problème de la synchronisation des horloges dans des réseaux filaires. Leur principal inconvénient est qu'ils sont couteux en énergie, ce qui n'est pas compatible avec le profil des WSNs. En ce qui concerne les solutions dédiées aux réseaux de capteurs sans fil, le problème c'est qu'elles ne répondent pas au besoin en terme de performance pour la précision de la synchronisation.

Ainsi, concevoir une solution de synchronisation hybride, qui tient compte du problème de l'énergie et de la complexité d'implémentation d'un protocole, peut être plus robuste et plus performant que les solutions déjà étudiées.

Pour diminuer le coût et l'effort investis pour assurer une synchronisation des horloges, la majorité des protocoles conçus pour les réseaux de capteurs, comme par exemple le SMACS (Self-Organizing Medium Access Control for Sensorsnets), ont tiré avantage du principe de fonctionnement du protocole TDMA [76]. L'idée est que le canal est divisé en intervalles de temps, qui forment une trame et qui se répètent cycliquement au fil du temps. Dans chaque trame, le nœud est assigné à un intervalle dans lequel il peut

exécuter « ses actions » ; l'ensemble de tous ces intervalles forme un calendrier. La spécificité de ce calendrier dans un WSN est qu'il peut être fixe (ça veut dire construit au départ par la station de base) ou hybride (cas dans lequel la structure varie au fil du temps en fonction du comportement des capteurs).

Enfin, le protocole TDMA qui est à la base un protocole de couche MAC, est une solution de synchronisation évolutive, mais limitée quand il s'agit d'une erreur de communication ou des pertes de paquets. Donc, exploiter uniquement cette piste nous paraissait peu fiable pour les exigences de notre application.

En essayant de valoriser un mode de fonctionnement spécifique pour les capteurs, le mode à plusieurs états, [95] propose une solution innovante de synchronisation basée sur le principe de la gravitation. L'idée des auteurs est de synchroniser en premier ces différents états (mode actif, sommeil, etc.) et ensuite les groupes de capteurs ainsi formés entre eux.

Si nous analysons maintenant notre réseau (Figure I.17), nous pouvons identifier également des groupes homogènes d'éléments (par exemple Routeur1 : Noeud11...Noeud18) qui pourront se synchroniser d'une manière individuelle, étape suivie par la suite d'une synchronisation globale du système.

Analysons maintenant le principe de fonctionnement du protocole LEACH (Low-Energy Adaptive Clustering Hierarchy). Il utilise le TDMA pour assurer la communication entre les nœuds et leur « cluster head », qui peuvent envoyer ainsi à la station de base les messages reçus de leurs nœuds rattachés [40]. L'hypothèse est que les nœuds du cluster commencent la phase de configuration dans le même temps et restent synchronisés par la suite. Le mécanisme qui assure la synchronisation globale du système est confié à la station de base qui envoie des impulsions de synchronisation à l'ensemble des nœuds [96].

Cette ultime hypothèse nous permet d'accomplir en intégralité, un nouvel algorithme de synchronisation pour de réseaux de capteurs sans fil. Ainsi, nous décidons d'utiliser dans notre topologie hiérarchique un modèle de couche MAC de type TDMA, pour les raisons discutées précédemment. De plus, l'organisation par groupes du réseau nous permet d'imaginer un modèle de synchronisation par niveaux : Concentrateur-Routeurs et Routeurs-Nœuds. Pour le premier niveau de communication nous déciderons d'appliquer l'algorithme de synchronisation qui caractérise la norme IEEE-1588 pour ses meilleures performances en terme de synchronisation. En ce qui concerne le Niveau 2 de synchronisation (routeurs-nœuds), nous nous proposons d'imaginer un algorithme hybride, qui combinerà le principe du protocole IEEE-1588, choisi pour sa précision et celui du protocole de synchronisation PBS, choisi pour ses avantages de point de vue économie d'énergie.

La présentation de notre solution pour résoudre le problème de la synchronisation dans un réseau de capteurs sans fil, est détaillée au chapitre III de ce manuscrit.

I.5.4 Le besoin d'économie d'énergie

Les techniques pour minimiser la consommation d'énergie présentées tout au long de ce chapitre sont nombreuses et peuvent être appliquées aux différents niveaux de communication (couche MAC, liaison de données, réseau, etc.). Dans la Figure I.18, nous

présentons les principaux axes que nous avons identifié dans la tentative de faire de la gestion d'énergie, problématique essentielle pour la démarche générale de concevoir et optimiser les réseaux de capteurs sans fil, et plus particulièrement, les réseaux d'instrumentation et de mesures.

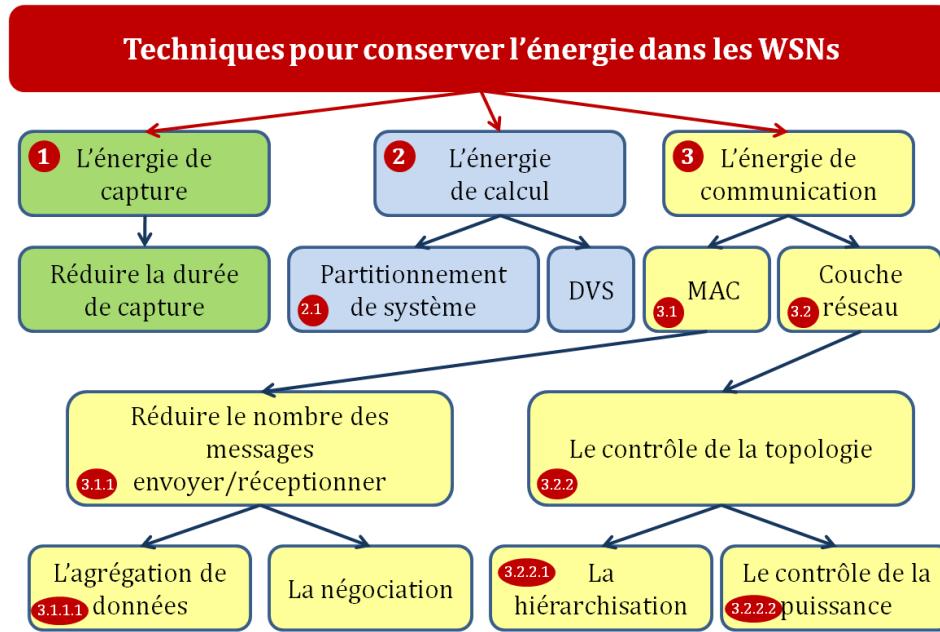


Figure I.18 Techniques pour minimiser la consommation d'énergie.

En suivant les trois axes principaux qui ressortent de la figure ci-dessus (les axes 1, 2 et 3 de la Figure I.18), nous essayons de proposer les solutions les plus adaptées pour notre système de communication. Comme il existe plusieurs grandes sources consommatrices en énergie, nous nous retrouvons également avec plusieurs possibilités pour faire de l'économie d'énergie. Le choix de la stratégie se fait, comme pour la synchronisation, en fonction du contexte et de l'application.

L'AXE NUMERO 1

Pour réduire la quantité d'énergie consommée dans le procédé de capture, la seule possibilité est de réduire la durée effective de capture, ça veut dire le temps pendant lequel le nœud capteur consomme le plus d'énergie. Dans notre application, les utilisateurs ont besoin d'une interaction continue avec l'environnement surveillé, donc l'activité de nœuds capteurs doit s'effectuer sans interruption, ce qui impose une détection permanente des informations. Autrement dit, nous ne pouvons pas réaliser une économie d'énergie de cette manière là.

L'AXE NUMERO 2

Une grande quantité d'énergie est consommée dans les calculs, et les solutions proposées dans ce sens ne sont pas nombreuses. Une des possibilités est de faire un partitionnement de système (point 2.1), c'est-à-dire transférer les calculs gourmands en énergie et de longue durée sur une station de base qui n'a pas de problèmes énergétiques. Notre architecture réseau dispose d'une station alimentée en continu, le concentrateur, qui va recevoir et traiter toutes les informations recueillies par les nœuds capteurs à travers les routeurs. Ainsi, les autres éléments du système (les routeurs et les capteurs) vont être

conçus d'une manière plus simple, ce qui va diminuer leur coût et augmenter leur durée de vie.

La deuxième solution consiste à utiliser l'approche DVS (Dynamique Voltage Scaling) [95], qui permet de diminuer la tension et par conséquence l'énergie consommée par un ordinateur portable ou d'autres appareils mobiles. A ce niveau, cette méthode reste envisageable pour notre système, mais en terme de perspective d'optimisation.

L'AXE NUMERO 3

L'énergie de communication entre les éléments du réseau reste la solution la plus exploitée dans le besoin de gestion d'énergie. L'explication est simple : elle est directement liée à l'architecture des protocoles de communication, qui représente la partie la plus complexe et en même temps la plus facile à exploiter pour faire de l'optimisation. Les couches les plus concernées sont : liaison de données et réseau.

Ainsi, les techniques utilisées au niveau MAC (point 3.1) sont basées sur l'agrégation de données, la négociation, les théories de fusion et de décision, dans le but commun de diminuer le nombre de messages envoyés et réceptionnés. C'est aussi le cas de notre système, où nous avons proposé une agrégation de données au niveau des routeurs (point 3.1.1.1), ce qui va diminuer le nombre de paquets échangés. Par conséquent, la communication va être plus fiable (nous aurons un paquet de donnée au lieu de « n », avec des mécanismes de détection et de correction d'erreurs) et l'énergie économisée au niveau de routeurs va prolonger la durée de vie du réseau. Si nous prenons l'hypothèse que les routeurs pourront être alimentés en continu, cette stratégie reste toujours efficace de point de vue fiabilité de la communication et cette fois-ci, conservation d'énergie pour l'ensemble de l'avion (où l'énergie disponible doit être très bien gérée car elle est aussi limitée).

Une stratégie efficace de niveau réseau (point 3.2) est le contrôle de la topologie, comme la hiérarchisation du réseau en plusieurs niveaux. Nous avons proposé pour notre système une topologie hiérarchique (point 3.2.2.1), qui fonctionne sur deux niveaux de communication.

Le contrôle de la puissance de transmission (point 3.2.2.2) joue aussi sur la durée de vie du réseau en diminuant la puissance d'émission, et donc les collisions avec d'autres nœuds qui sont aussi en train de transmettre. Nous avons l'intention d'appliquer cette stratégie dans une proposition qui consiste à concevoir et utiliser un modèle d'antennes directives, compatible avec les contraintes de notre application. Ce qui caractérise ces antennes est leur capacité de focaliser dans une/plusieurs direction(s) précises. Cette propriété apporte de nombreux avantages qui vont être détaillés dans le chapitre IV de ce mémoire.

Une autre possibilité pour faire de l'économie d'énergie est d'introduire, si l'application le permet, plusieurs états de fonctionnement pour les capteurs, ce qui n'est pas le cas de la nôtre (car les capteurs doivent rester actifs en permanence).

Encore une fois, nous pouvons constater que la littérature ne propose pas une solution complète et adaptée pour tous les besoins de réseaux de capteurs sans fil, et encore moins pour les réseaux d'instrumentation et de mesures. Les fonctions qui consomment le plus d'énergie dans un nœud, notamment la capture, le traitement et la communication, sont influencées par des facteurs comme le médium de communication radio, la méthode d'accès au canal, le routage, etc. Par conséquent, pour pouvoir trouver une solution de conservation

d'énergie fiable, il faudra tenir compte de tous ces facteurs dans la stratégie de conception d'une architecture réseau.

I.6 Conclusion

Dans ce chapitre nous nous sommes concentrés sur la présentation de réseaux de capteurs sans fil, une catégorie de systèmes qui vont faire partie intégrante de notre vie quotidienne, grâce à leur grande capacité de détection, leur flexibilité et leur adaptabilité dans les contextes les plus variés. Les applications qui tirent profit de leurs avantages réunissent des nombreux domaines, comme la santé, la météorologie, le militaire, l'environnemental, la sécurité, etc.

Par conséquent, pour faire face à toutes les contraintes de ces applications, la mise en œuvre des réseaux de capteurs sans fil nécessite tout d'abord une étude préliminaire des technologies et des protocoles existants, une phase de vérification par simulation, suivie de la phase finale de réalisation. Le système ainsi conçu doit satisfaire des besoins génériques comme la tolérance aux pannes, l'économie d'énergie, la scalabilité, le coût réduit etc., où des besoins spécifiques à l'application comme une grande précision pour la synchronisation des données, la gestion d'un nombre élevé de capteurs, le choix du matériel pour fonctionner dans des conditions extrêmes (comme par exemple à de très grandes variations de température et de pression), etc.

Dans ce cadre général des réseaux de capteurs sans fil, nous nous sommes intéressés plus particulièrement aux réseaux destinés à l'instrumentation et aux mesures. Ces réseaux de communication sont spécialisés dans l'automatisation des équipements pour dialoguer avec les instruments et les systèmes de mesures tels que les logiciels de configuration, de réglages, de calibration et de diagnostic. Le grand apport de ces réseaux est de pouvoir automatiser et optimiser la maintenance et le contrôle de différentes structures, dans de multiples domaines.

Notre travail se place dans le contexte de l'application SACER. La problématique consiste à concevoir une architecture de communication sans fil pour un système de multi capteurs, qui doit être autonome et fiable. Les principaux défis à résoudre sont : la robustesse du réseau avec un routage efficace, une très bonne précision pour la synchronisation des données et l'efficacité de la gestion d'énergie.

La conclusion que nous pouvons tirer de toutes les études présentées précédemment est qu'une solution idéale n'existe pas. La stratégie est donc d'analyser les contraintes et les besoins de l'application qui vont nous fournir des informations sur les technologies les mieux adaptées et qui vont assurer un bon fonctionnement du système.

Concernant le problème de la synchronisation, il peut être résolu soit à travers un mécanisme de gestion, fruit de collaboration de plusieurs niveaux de communication, soit à travers une application dédiée, mais qui doit prendre en compte l'aspect économie d'énergie, ce qui n'est pas facile. Nous avons décidé de proposer une solution qui utilise les deux mécanismes, dans le but de pouvoir assurer une meilleure précision de la synchronisation et en même temps de réaliser des économies d'énergie.

En ce qui concerne l'aspect énergétique, nous avons fait le choix de plusieurs stratégies de gestion possibles (notamment pour l'énergie de calcul et de communication) ; cela va nous permettre de prolonger au maximum la durée de vie de notre système.

Dans la suite de ce mémoire (chapitre II), nous présenterons les détails de notre application et de parcourir les étapes de conception, qui nous seront nécessaires pour la mise en œuvre et l'optimisation finale du système. Dans le chapitre III, nous proposons un nouveau protocole qui aura comme but de garantir une bonne précision de la synchronisation et également de tenir compte du besoin pour la minimisation de la consommation d'énergie. Le chapitre IV va traiter des aspects liés au fonctionnement et aux avantages offerts par l'utilisation des antennes directives, notamment sur l'aspect consommation d'énergie, ainsi que la qualité de service. Le dernier chapitre (chapitre V) aura pour objectif de présenter tous les résultats de simulations et d'expérimentations que nous avons réalisées pour la conception de notre système de communication.

II CONCEPTION ET MODELISATION D'UN RESEAU D'INSTRUMENTATION ET DE MESURES SANS FIL

| | | |
|-------------|--|-----------|
| II.1 | Introduction | 53 |
| II.2 | Description de notre réseau d'instrumentation et de mesures | 55 |
| II.2.1 | Topologie du réseau | 55 |
| II.2.2 | Méthodologie et choix de technologies | 56 |
| II.2.3 | Spécification des besoins | 58 |
| II.2.4 | Analyse préliminaire du système | 58 |
| II.2.5 | Conception du système | 63 |
| II.3 | Description de la plateforme matérielle | 72 |
| II.3.1 | Carte Wisair Wireless USB PCI Express Mini | 72 |
| II.3.2 | Carte NI PCI-1588 | 73 |
| II.3.3 | Plateforme ML410 | 74 |
| II.4 | Modélisation sur NS-2 | 75 |
| II.4.1 | Eléments de base de la simulation en NS-2 | 76 |
| II.4.2 | Les principales contributions en NS-2 | 78 |
| II.5 | Conclusion | 81 |

Ce deuxième chapitre débute par la présentation d'une application d'instrumentation et de mesures, qui va servir comme support d'étude et d'analyse dans les prochains chapitres. Après avoir introduit la problématique des WSNs dans le chapitre I, nous allons présenter la topologie, le fonctionnement et les contraintes liées à notre application. Par la suite, nous allons nous concentrer sur le choix des technologies et des protocoles les mieux adaptés dans ce contexte, pour enfin pouvoir répondre aux besoins et proposer des solutions améliorées. Ces propositions vont être ensuite implémentées dans l'outil de simulation NS-2 pour les valider.

II.1 Introduction

Il ne s'agit pas dans cette introduction d'une présentation détaillée des réseaux d'instrumentation et de mesures, mais plutôt d'un rappel des principaux concepts indispensables pour comprendre la problématique traitée dans ce manuscrit.

Le concept de réseau de capteurs sans fil pour l'instrumentation et la mesure est à la base un ensemble de sources d'informations qui sont chargées de contrôler et d'enregistrer des activités, de surveiller leur évolution et de signaler une éventuelle dégradation de leur état. La diversité des tâches à accomplir confère à ces réseaux leur caractère bien particulier.

Les domaines d'application sont divers et vont du simple configurateur local doté de fonctionnalités similaires à celles d'un bus de terrain: surveillance et gestion des équipements, transmissions des données de mesures, suivi des équipements mobiles, vidéosurveillance, jusqu'à des besoins de transmission à 6000 Km à travers la technologie GPRS. Le point commun de toutes ces applications est l'absence de fil, qui donne plus de flexibilité et enlève la surcharge due à la masse du câble. Leur but commun est de ramener les informations d'un instrument situé sur le terrain vers l'unité centrale, responsable de la commande et de la configuration.

Tous ces types d'applications demandent également un caractère autonome du capteur d'un point de vue alimentation. Cela se traduit par deux contraintes supplémentaires: pouvoir minimiser la consommation du nœud capteur et arriver à maximiser la durée de vie de la batterie.

Ces réseaux de communication sans fil se caractérisent généralement par un faible débit, une portée radio inférieure à quelques centaines de mètres et une consommation en énergie qui dépend de la consommation du capteur et de sa fréquence de scrutation³⁰. Ici, nous devons préciser qu'il existe des équipements qui consomment peu (comme les capteurs de températures, de pression, de niveau, de position, etc.) et d'autres équipements comme un capteur de niveau par ultrason, ou un débitmètre électromagnétique, qui sont plus gourmands en énergie.

Ainsi, les protocoles de communication doivent être capables d'améliorer cette limitation énergétique des capteurs, par des mécanismes d'accès au medium, de routage efficace et des applications qui prennent en compte implicitement cet aspect.

Nous avons vu que la portée radio n'est pas excessive mais qu'elle est aussi influencée par le problème énergétique. Les antennes doivent avoir un rayonnement qui, souvent, ne couvrira pas l'étendue du réseau pour ne pas entraîner trop de consommation d'énergie. Cela amènera parfois les nœuds du réseau à échanger des paquets en multi-sauts. Mais, même si la puissance de transmission est réduite, le module de transmission/réception reste l'élément le plus consommateur du capteur. Pour cela, des solutions pour réaliser une économie d'énergie existent et sont des sujets de recherche actifs.

Une autre caractéristique importante pour un système d'instrumentation et de mesure est la synchronisation de données véhiculées dans le réseau, qui est due au mode de fonctionnement en temps réel. Ce fonctionnement va avoir des conséquences également en ce qui concerne le débit, qui va être plus élevé par rapport celui d'un réseau de capteurs sans fil.

En ce qui concerne la topologie des réseaux de capteurs sans fil, plusieurs possibilités sont envisageables: le mode ad hoc [97] ou le mode avec des points d'accès qui rappelle une structure en cellules [98]. Le choix doit être adapté et est fortement lié aux contraintes de chaque type d'application.

Après avoir rappelé les caractéristiques générales de ces réseaux, une description détaillée de l'application associée à notre étude s'impose, afin de pouvoir trouver les solutions techniques qui soient adaptées.

³⁰ Ce paramètre détermine les intervalles auxquels l'OS récupère/rafraîchit les données.

II.2 Description de notre réseau d'instrumentation et de mesures

Dans l'optique de concevoir un système de communication pour les réseaux d'instrumentation sans fil, nous nous baserons sur les besoins d'une telle application.

C'est le cas du projet SACER (Système Autonome Communicant Embarqué en Réseau) qui a pour but principal de mettre au point un réseau de capteurs sans fil de pression et de température. Plus précisément, il s'agit de concevoir un système autonome multi capteurs, d'acquisition et de traitement avec mémorisation et communiquer à travers un réseau sans fil pour la mesure embarquée, adapté aux contraintes aéronautiques.

Pendant la période des essais en vol, les avions sont aujourd'hui instrumentés par un ensemble de capteurs de pression et de température filaires permettant d'établir une cartographie des écoulements d'air sur les ailes. Ces capteurs doivent être packagés dans des sortes de « gants » collés à l'aile et reliés à la cabine par des torons de câbles. Ainsi les ailes sont percées afin de faire circuler ces câbles, avec tous les risques de modification de la structure que cela entraîne. Les contraintes du projet SACER découlent principalement de cet application, c'est à dire un faible encombrement (moins de 2mm d'épaisseur), une résistance à des températures extrêmes, etc.

II.2.1 Topologie du réseau

Nous rappelons que le système est constitué de trois éléments de réseau interconnectés (Figure II.1):

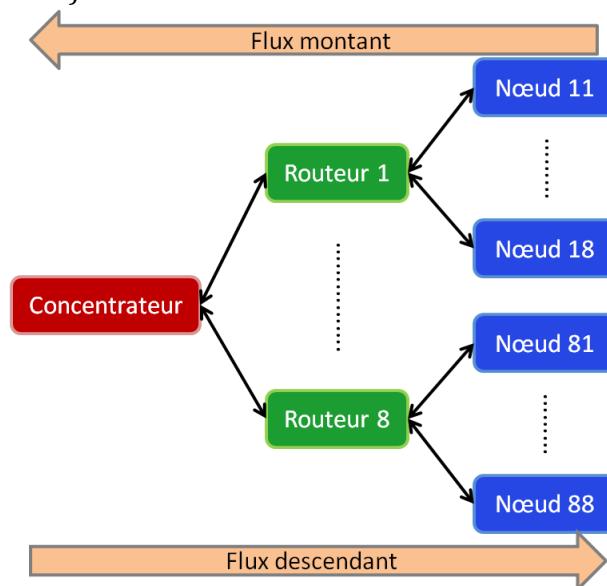


Figure II.1 Les flux de communication dans la topologie SACER.

- ❖ **Le concentrateur** – l'élément qui fait le lien entre le réseau SACER et le monde extérieur. Il concentre et distribue les communications de données, connaît l'intégralité du réseau et le commande. En effet, il réalise le lien entre toutes les données acquises et le réseau ou l'application du client. Il est également en charge de la gestion du réseau;

- ❖ **Le routeur** – l'élément intermédiaire assurant le routage des données dans le réseau. C'est un répéteur plus ou moins passif des données acquises qui a pour fonction de relayer l'information montante (nœud → concentrateur) et descendante (concentrateur → nœud) ;
 - ❖ **Le nœud** – la chaîne d'acquisition de données depuis les capteurs, il comporte généralement un ou plusieurs élément(s) de capture(s).
- Pour ce qui est des flux de communication, nous allons considérer deux types de flux, illustrés dans la Figure II.1 :
- ❖ **Le flux descendant** correspond à toutes les données qui sont émises par le concentrateur et qui transitent à destination d'un ou des nœuds via les routeurs.
 - ❖ **Le flux montant** correspond à toutes les données qui sont émises par les nœuds et qui transitent à destination du concentrateur via les routeurs.

II.2.2 Méthodologie et choix de technologies

Notre application a pour but de mettre au point un réseau de centaines de capteurs qui communiquent sans fil. De fait, certains de nos travaux s'appuient sur le travail réalisé par les autres partenaires du projet SACER, en utilisant leur matériel, ou par la modélisation par nos soins de certaines briques dans le logiciel NS-2.

Comme nous avons vu précédemment, il existe de nombreuses contraintes qui doivent être résolues par la coopération entre différentes disciplines, correspondant aux spécifications des partenaires du projet:

1. Le développement des applications et des interfaces graphiques

Cette partie consiste à l'intégration de la couche application de données et à la simulation globale du réseau, qui fait l'objet du dernier chapitre (chapitre V). Issues des besoins applicatifs, notre contribution a consisté à modéliser ces flux sous NS_2.

2. La conception des antennes planaires UWB

Ces antennes développées par les partenaires électroniciens du LAAS, seront utilisées dans la maquette pour être connectées aux capteurs et puis testées avec des cartes Wisair Wireless USB PCI Express Mini³¹ qui utilisent la technologie de transmission UWB. Les détails techniques et les résultats seront présentés également dans le chapitre V.

3. Le développement de la couche physique et MAC

Cette étape implique la conception de la couche PHY ainsi que de la couche MAC, avec les organes de transmission tels que les antennes émettrices. L'intégration d'un modèle d'antennes directives dans le logiciel NS-2 est également prévue, travail qui est détaillé dans le chapitre IV. Les couches basses (PHY et MAC) ont été proposées par les électroniciens du LAAS [99], nous avons ainsi utilisé leurs résultats pour développer un modèle de simulation afin de tester les couches hautes.

4. Le développement de la couche réseau

La conception de la couche réseau doit se faire en complémentarité avec les couches basses, ce qui est indispensable afin de bien concevoir l'échange des paquets.

5. La proposition d'une solution pour la synchronisation

La précision demandée pour la synchronisation des données est de 1 µs - 500 ns. Ce travail implique tous d'abord une étape de test du matériel existant, i.e. des cartes NI PCI-

³¹ <http://www.wisair.com/wp-content/PCIe-Mini-Card.pdf>

1588³² basées sur la norme IEEE 1588, et par la suite le choix/conception de la solution de synchronisation la mieux adaptée.

Nous reviendrons donc sur la proposition effectuée dans le chapitre III ainsi que sur ses performances dans le chapitre V.

6. La récupération et le stockage de l'énergie

Cette étape est consacrée à l'étude de l'utilisation des panneaux photovoltaïques couplés à des accumulateurs pour stocker et fournir l'énergie aux capteurs.

7. La réalisation des capteurs de pression et de température

Le but de ces études et recherches sont menées sur le meilleur compromis entre miniaturisation, autonomie et précision de mesure.

8. La conception des maquettes et des prototypes

Chaque partie prenante de ce système est conçue et testée sur une plateforme de développement, qui doit fonctionner conjointement dans le but final d'une réalisation d'un prototype.

Les points énumérés ci-dessus sur lesquels nous sommes intervenus tout au long de notre travail sont principalement les cinq premiers. Nous pouvons constater que les domaines de compétences sont multiples, mais aussi le type d'activités à accomplir: spécification, test, conception, implémentation, validation. Ainsi, le tableau suivant donne un aperçu du cadre de travail à réaliser :

| Domaines | Couches | Plateformes | Langages | Types d'activités |
|-------------------------------------|------------------------------|---|-----------------------|--|
| Electronique, Réseau & Informatique | Physique, MAC | Plateforme de tests, Carte Wisair Wireless USB PCI Express Mini, Antenne planaire | C, NS-2 | Tests & Mesures, Implémentation & Simulation |
| Réseau & Informatique industrielle | Réseau & Transport | FPGA ML410, PowerPC, Plateforme de tests | UML, VHDL/C NS-2, C++ | Spécifications, Conception, Implémentation, Validation |
| Electronique, Réseau & Informatique | Application /Synchronisation | Plateforme de tests, Carte NI PCI-1588, Carte Wisair Wireless USB PCI Express Mini, Antenne planaire, Oscilloscope LeCroy WavePro ³³ | NS-2, C++ | Tests & Mesures, Simulation, Validation |

Tableau II.1 L'ensemble du travail à réaliser.

La gamme de fréquence de communication choisie pour le projet se situe dans l'UWB (Ultra Wide Band) allant de 6 à 8,5 GHz, gamme autorisée en Europe pour ne pas interférer notamment avec les communications militaires (aux Etats-Unis, elle se situe entre 3,1 et 10,6 GHz).

³² http://www.ni.com/pdf/products/us/cat_pci1588.pdf

³³ http://www.lecroy.com/files/pdf/LeCroy_WavePro_7_Zi_Datasheet.pdf

Le projet est divisé en nombreuses sous parties, comme la gestion de l'énergie, la conception système, la réalisation d'éléments sensibles et enfin la transmission sans fil des mesures effectuées. L'objectif de ce dernier point est, plus précisément, d'implémenter un protocole de communication qui doit assurer la gestion et le transport des mesures et des données échangées entre les différents éléments du réseau (concentrateur, routeurs, nœuds). C'est ce point qui représente la partie la plus consistante de notre travail, où se rajoute la problématique de la synchronisation et de gestion d'énergie.

II.2.3 Spécification des besoins

Pour développer une application informatique quelconque, il est primordial de cibler les besoins des futurs utilisateurs dès le départ, étape identifiée lors de la phase de spécification. En effet, cette étape consiste à effectuer une analyse détaillée des exigences, qui sont nécessaires pour le développement de l'architecture à concevoir.

Pour notre application, les besoins des utilisateurs sont nombreux, ce qui nous a permis d'identifier les éléments suivants :

- ❖ Définir la pile des protocoles de communication et leur implémentations ;
- ❖ Assurer la communication entre les différents éléments du réseau (concentrateur, routeurs, nœuds) ;
- ❖ Réduire le taux de perte des données échangées ;
- ❖ Mettre en œuvre un système capable de gérer la qualité de service du réseau ;
- ❖ Proposer des mécanismes pour économiser l'énergie ;
- ❖ Garantir une bonne synchronisation de la totalité du système ;
- ❖ Concevoir une architecture simple et flexible.

Tous ces éléments doivent être pris en compte dans l'architecture globale du système, soit comme des mécanismes intégrés dans la pile de communication, soit en tant que protocoles définis pour résoudre une tâche spécifique.

II.2.4 Analyse préliminaire du système

Notre système de communication est un réseau « fixe » pour lequel le nombre de capteur est connu. Il est donc possible d'optimiser les protocoles et cette optimisation passe par des échanges. Par conséquent, nous avons pour charge de résoudre tous les points durs qui représentent des aspects de recherche.

La première étape dans le processus de développement des applications est la phase d'analyse, qui permet d'identifier les principaux besoins opérationnels et fonctionnels du système. Cette analyse consiste aussi à définir les acteurs ainsi qu'à spécifier les différents cas d'utilisation.

Notre but est donc de décrire les fonctions détaillées du système SACER avec des diagrammes des cas d'utilisation d'UML, à l'aide de l'outil open source TOPCASED³⁴. Cette technique permet de décrire sous la forme d'actions et de réactions le comportement d'un système, du point de vue utilisateur. Elle apporte une solution au problème de la détermination et de la compréhension des besoins [100].

³⁴ <http://www.topcased.org/>

Nous présenterons par la suite, plusieurs éléments nécessaires pour l'élaboration de notre application à l'aide d'un diagramme des cas d'utilisation [101].

Ainsi, les acteurs qui interviennent dans notre système (comme représenté dans la Figure II.2) sont:

- ❖ Le nœud – est l'élément communicant sans fil situé en bout de réseau, relié à un ou plusieurs éléments sensibles, qui peut communiquer avec un routeur ou le concentrateur s'il est à portée ;
- ❖ Le routeur – est l'élément communicant sans fil vers les nœuds et vers les autres routeurs ou le concentrateur. Son rôle est de faire le relais des informations qui transitent sur le réseau ;
- ❖ Le concentrateur – est l'élément communicant sans fil vers les autres éléments du réseau. Il est chargé de faire la passerelle entre le réseau SACER et l'application industrielle.

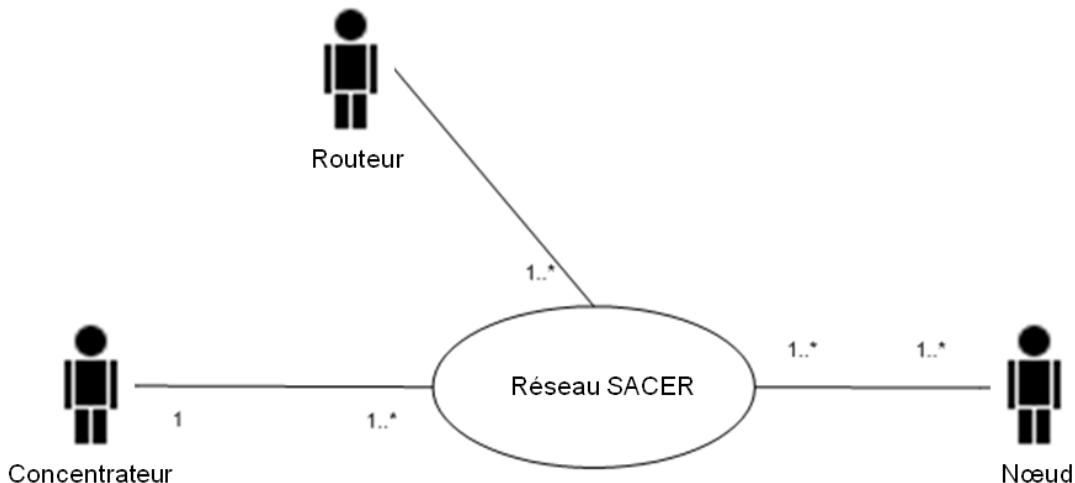


Figure II.2 Les acteurs de notre système.

Il faut noter que le flux descendant défini auparavant correspond au transfert des commandes depuis le concentrateur vers les nœuds et que le flux montant correspond au transfert des données depuis les nœuds vers le concentrateur. De plus, nous notons que la topologie du réseau est fixe, car les émetteurs ne se déplacent pas les uns par rapport aux autres. Nous remarquons aussi qu'il y a un point central unique (le concentrateur) qui communique avec des points intermédiaires (les routeurs) qui eux mêmes communiquent avec d'autres routeurs ou avec les éléments terminaux (nœuds).

Ceci étant rappelé, nous allons passer maintenant à la définition des cas d'utilisation de notre application. Du fait de la complexité de l'application, les définitions des cas d'utilisation seront données selon les acteurs intervenants.

LE DIAGRAMME DE CAS D'UTILISATION RELATIF AU CONCENTRATEUR

Dans le Tableau II.2, nous réalisons une description des principales actions effectuées par le concentrateur, comme le montre la Figure II.3.

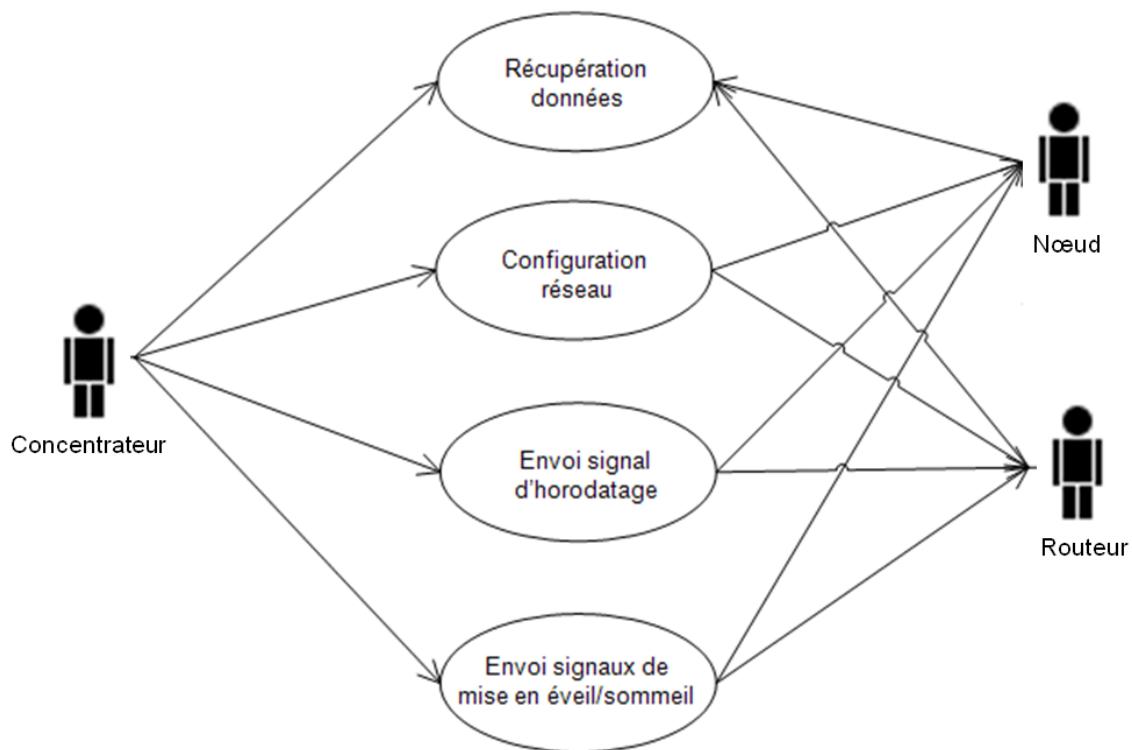


Figure II.3 Le diagramme du cas d'utilisation relatif au concentrateur.

| Action | Description |
|--|---|
| Récupération données | Récupération des données mesurées par les nœuds et non encore acheminées vers le concentrateur. |
| Configuration réseau | Distribution des ordres à l'ensemble du réseau pour adopter la configuration voulue. |
| Envoi signal d'horodatage | Envoi périodique d'un signal de synchronisation à tout le réseau (nœuds, routeurs). |
| Envoi signaux de mise en éveil/sommeil | Envoi des ordres de mise en éveil ou sommeil des différents éléments du réseau (nœuds, routeurs). |

Tableau II.2 Description du cas d'utilisation relatif au concentrateur.

LE DIAGRAMME DE CAS D'UTILISATION RELATIF AU ROUTEUR

Comme dans le cas du concentrateur, nous allons maintenant analyser les actions effectuées par le routeur de la même façon. Ainsi, le Tableau II.3 donne une description de toutes actions effectuées par un routeur (Figure II.4).

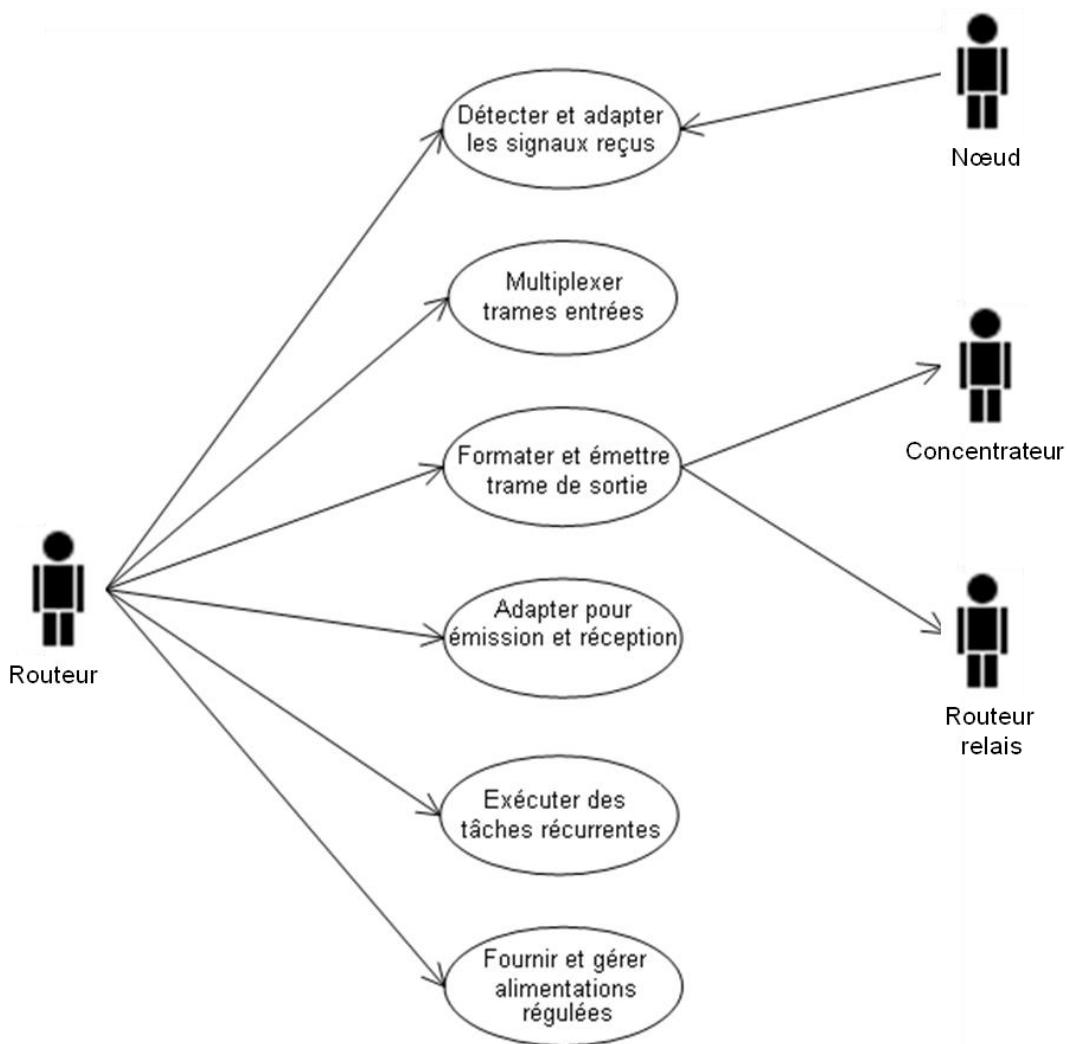


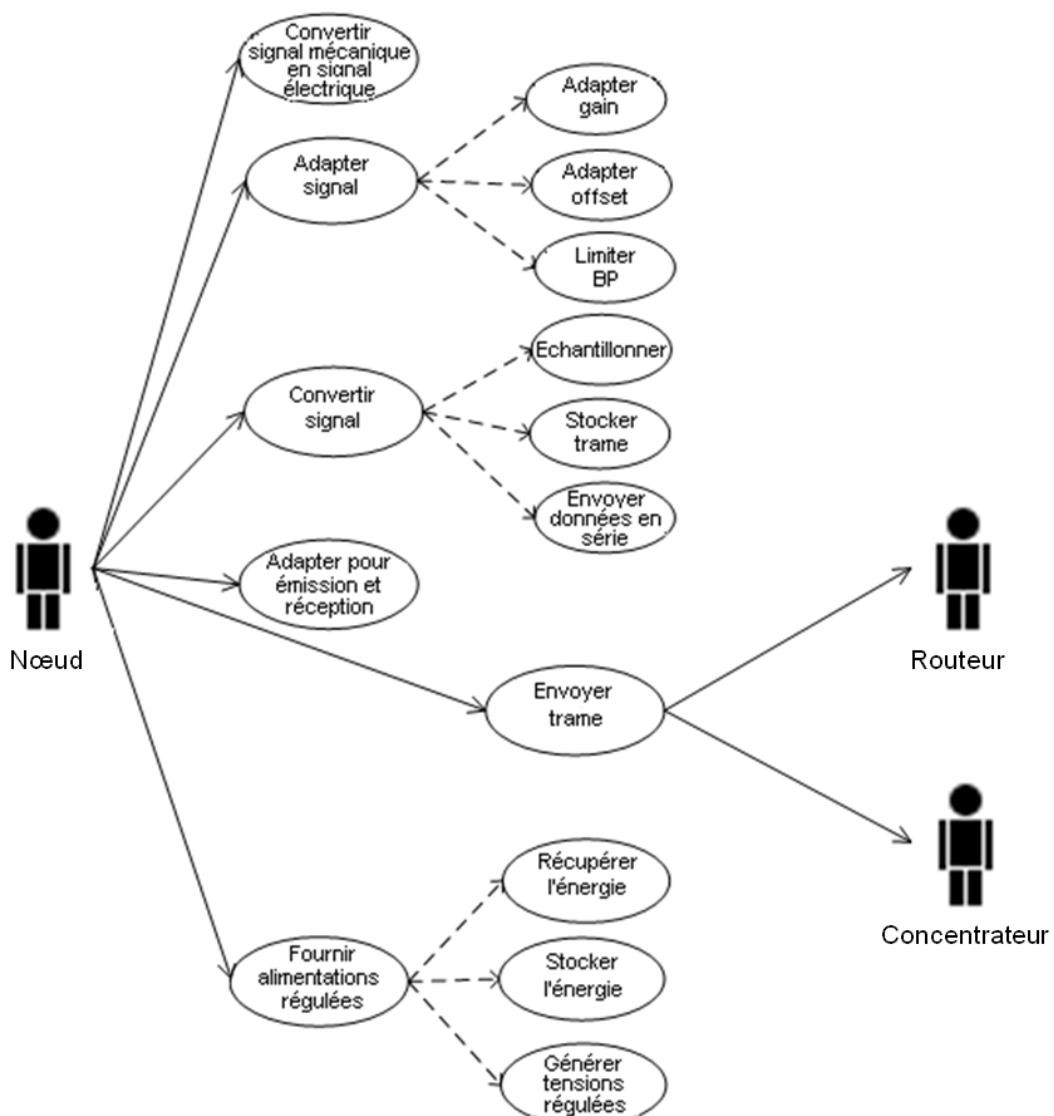
Figure II.4 Le diagramme du cas d'utilisation relatif au routeur.

| Titre | Description |
|--|---|
| Déetecter et adapter des signaux reçus | Réception des trames envoyées par des nœuds ou bien un autre routeur. |
| Multiplexer les trames d'entrées | Mettre en forme le multiplexage des différentes trames collectées par le routeur afin d'obtenir une trame unique encapsulée. |
| Format et émettre trame de sortie | La trame issue du multiplexage est envoyée vers un autre routeur (relais) ou bien directement vers le concentrateur. |
| Adapter pour émission et réception | Les interfaces radio fréquence-signal en bande de base doivent être configurées (puissance d'émission, débit de transmission, type de modulation) d'après la configuration réseau et la nature du lien radio sollicité. |
| Exécuter des tâches récurrentes | Gérer les tâches récurrentes que ce soit la |

| | |
|---|---|
| | synchronisation (tâche périodique) ou la demande d'un renvoi de trame après la détection de données erronées (tâche apériodique), la gestion de l'état, la configuration. |
| Fournir et gérer l'alimentation régulée | Fournir des tensions régulées aux circuits du routeur et signaler une anomalie d'alimentation au concentrateur. |

Tableau II.3 Description du cas d'utilisation relatif au routeur.**LE DIAGRAMME DE CAS D'UTILISATION RELATIF AU NŒUD**

Le Tableau II.4 décrit les principales actions effectuées par un nœud du réseau, et illustrées dans le diagramme de la Figure II.5 :

**Figure II.5 Diagramme du cas d'utilisation relatif au nœud.**

| Titre | Description |
|--|--|
| Convertir le signal mécanique en signal électrique | Délivrer une tension proportionnelle à la grandeur mesurée. |
| Adapter signal | Adapter la dynamique du signal au convertisseur Analogique/Numérique. |
| Adapter gain | Adapter le niveau du signal analogique afin d'obtenir une dynamique de pleine échelle maximale sans effet de saturation. |
| Adapter offset (décalage) | Compenser les décalages des éléments sensibles ou de l'électronique afin d'obtenir une dynamique de pleine échelle maximale. |
| Limiter BP (Bande Passante) | Filtrage anti-repliement de spectre du signal analogique avant la conversion numérique. |
| Convertir signal | Convertir un signal analogique en grandeur numérique codée sur 16 bits. |
| Echantillonner | Echantillonnage – blocage du signal analogique et conversion A/D (Analogique/Digital). |
| Stocker trame | Stocker temporairement les données numériques. |
| Envoyer données en série | Mettre en série les données numériques afin de pouvoir les transporter sur un lien radio. |
| Adapter pour émission et réception | Paramétrier l'interface radio fréquence-signal en bande de base selon le lien radio sollicité. |
| Envoyer trame | Envoyer une trame vers un routeur/concentrateur. |
| Fournir alimentations régulées | Délivrer des tensions stables et détecter une anomalie sur les alimentations. |
| Récupérer l'énergie | Récupérer de l'énergie disponible dans l'environnement physique immédiat et la convertir en énergie électrique. |
| Stocker l'énergie | Stocker une partie de l'énergie récupérable afin de stabiliser l'offre d'énergie dans le temps. |
| Générer tensions régulées | Délivrer sous forme d'une tension stable et contrôlée de l'énergie électrique aux circuits du nœud. |

Tableau II.4 Description du cas d'utilisation relatif au nœud.

Pour conclure, nous avons finalisé l'étape d'analyse préliminaire du système, qui a consisté à identifier les principaux éléments intervenants dans cette architecture de communication, ainsi que les cas d'utilisations qui en découlent. Nous constatons que le nombre d'actions qui caractérisent les nœuds est plus significatif par rapport à celui de routeurs ou de concentrateur. Cette différence reflète le travail important qui doit être réalisé au niveau des nœuds dans le réseau. Ceci va nous permettre de passer à l'étape suivante, celle de conception du système.

II.2.5 Conception du système

La phase de conception système consiste à définir l'ensemble des éléments fonctionnels et opérationnels qui constituent l'application. Nous commencerons par la conception des différents formats des trames échangées par le réseau, et par la suite, nous présenterons les différents modules intervenant dans notre architecture.

Ainsi, dans la Figure II.6 sont illustrées les grandes étapes issues de la simplification des cas d'utilisation discutés auparavant.

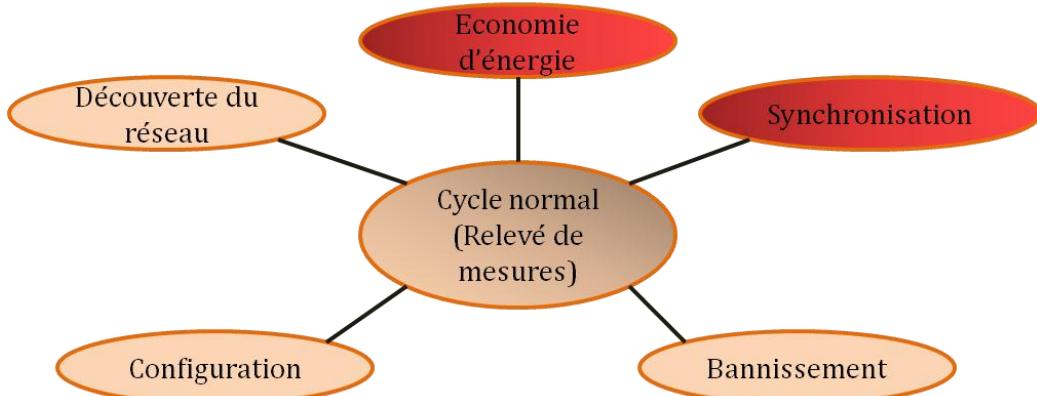


Figure II.6 Les éléments du processus de communication.

Ainsi, le cycle normal de communication s'identifie avec la phase de demande et de remontée des valeurs échangées entre les éléments du réseau. Autour de ce cycle, gravitent les autres différents cas d'utilisation :

- ❖ La découverte du réseau – correspond à l'étape de communication où tous les éléments du réseau s'identifient les uns par rapport aux autres;
- ❖ La configuration – est une phase qui peut être exécutée dans les deux sens de communication, qui correspond à une demande explicite d'information depuis la supervision (la cabine d'avion);
- ❖ Le bannissement – ne caractérise que le flux descendant de communication et correspond à la désactivation d'un nœud ;
- ❖ La synchronisation – correspond au mécanisme qui assure la précision de la synchronisation ;
- ❖ L'économie d'énergie – caractérise les stratégies prises en compte pour faire de la gestion d'énergie.

Par la suite, nous présenterons les caractéristiques et la structure des trames pour chaque niveau de communication, ainsi que pour les cas d'utilisation que nous avons présenté ci-dessus.

II.2.5.1 Description des couches liaison de données et physique

LA COUCHE LIAISON DE DONNEES

Les équipements s'organisent selon un arbre de communication, comme représenté dans la Figure II.7. Il existe deux types d'arbres. Le premier possédant une unique racine et de 1 à 8 éléments de Niveau 1. Le second possède une unique racine, un unique élément de Niveau 1 et facultativement jusqu'à 8 éléments de Niveau 2.

Chaque élément se comporte comme un esclave vis-à-vis de son père et comme un maître vis-à-vis de ses fils. Un arbre possède deux flux de communication. Un flux descendant

depuis les extrémités de l'arbre vers la racine, un flux descendant depuis la racine dirigé vers les extrémités. Un élément ne peut communiquer qu'avec son maître et ses esclaves. Nous pouvons d'ores et déjà remarquer qu'il n'y aura pas de différence de comportement entre un élément de Niveau 1 sur un arbre de type 1 et un élément de Niveau 2 sur un arbre de type 2.

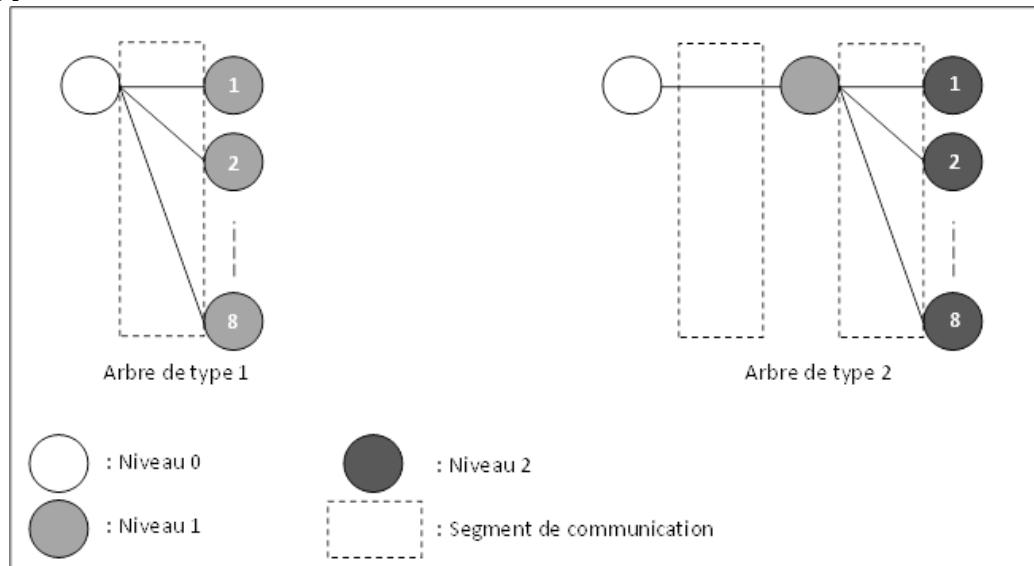


Figure II.7 Topologie de niveau MAC Concentrateur-Routeur-Nœud.

Le médium sans fil est organisé en 2 segments de communication. Un segment est partagé entre un maître et son ou ses esclaves et possède le même niveau que le maître. Il possède une bande de fréquence qui lui est allouée de manière à ce qu'elle soit suffisamment éloignée des fréquences utilisées par le segment voisin ou encore par les segments des liaisons voisines. Sur chaque segment, l'accès au médium est organisé par un multiplexage temporel (TDMA). Un TDMA est composé de 9 slots de tailles égales dont le premier est réservé au flux descendant (le maître communique avec ses esclaves) et les huit autres aux flux montant (les esclaves répondent au maître). Une super-trame est composée d'une trame descendante et de 8 trames montantes, illustrée sur la Figure II.8.

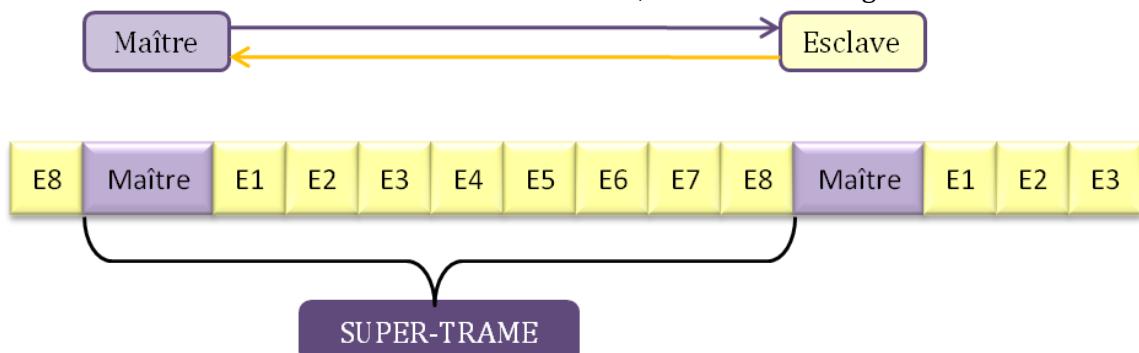


Figure II.8 Organisation d'une super-trame de la liaison de données.

Nous notons que ce cycle de communication est continu, quelque soit le cas d'utilisation. Il faut aussi préciser que les PDU³⁵ Réseaux seront transportés dans ces trames MAC et plus précisément, dans la partie données de ces trames via l'encapsulation.

³⁵Le PDU représente l'ensemble des informations échangées entre les différents niveaux de couches du Modèle OSI.

Dans le cas d'un arbre de type 2, l'élément de Niveau 1 se comporte comme un répéteur (routeur). L'élément de Niveau 1 pourra ne pas répéter la trame de l'un de ses esclaves afin de transmettre sa propre trame. La donnée de l'esclave choisi sera (selon implémentation) soit perdue, soit stockée par le maître pour être retransmise ultérieurement. Le temps d'une super-trame ne devra pas excéder la période d'échantillonnage des capteurs qui est de $30,5\mu\text{s}$ et le temps d'une trame le 9^{ème} de cette période: $3,39\mu\text{s}$. Une bonne synchronisation est donc essentielle au bon fonctionnement du TDMA.

Le format de PDU adopté par cette couche liaison de données a une taille fixe de 192 bits. Chaque trame devant être émise en $3.39\mu\text{s}$, ceci nous amène à un débit binaire au niveau de la couche liaison de 56.64 Mbits/s (Figure II.9).

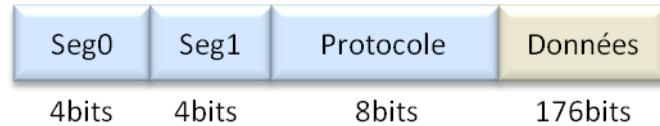


Figure II.9 Format du PDU de la liaison de données.

Les champs **Seg0** et **Seg1** permettent aux différents éléments de se repérer dans le TDMA et ainsi de contrôler leur accès au médium. Pour permettre une pile protocolaire modulable et complète, il a été nécessaire d'ajouter au format initial un champ **Protocole**, définissant le point d'accès au service entre la couche MAC et les couches sous-jacentes. Ce champ ne sera codé que sur 8 bit, il permettra ainsi 256 points d'accès au service. Dans le protocole Ethernet, le champ **protocole** est sur 2 octets et permet un plus large multiplexage. Enfin le champ **Données** permet le transport des PDU des couches supérieures.

Au niveau de la couche MAC, l'échange des trames se fait comme l'indique la Figure II.10:

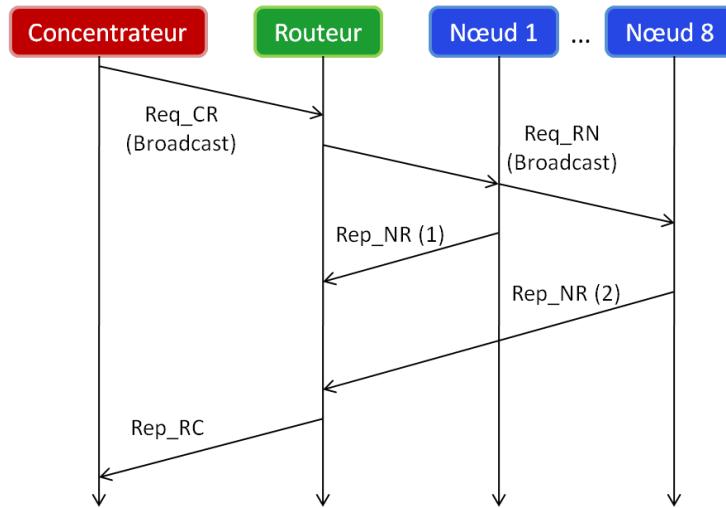


Figure II.10 Diagramme de séquence d'échange des trames d'un cycle normal.

Pendant cette phase, le concentrateur envoie une requête de type **Req_CR** en broadcast vers tous les routeurs. Chaque routeur à son tour, envoie une requête de type **Req_RN** en broadcast vers tous les noeuds qui lui sont associés. Ces derniers vont répondre par une trame réponse de type **Rep_NR**. Par conséquent, les routeurs vont répondre au concentrateur par une trame réponse de type **Rep_RC**.

LA COUCHE PHYSIQUE

Le FEC est un mécanisme de contrôle d'erreurs, qui se base sur un code capable de détecter et de corriger les erreurs. Cette correction des données se fait au niveau du récepteur sans le recours à la retransmission des données. La solution choisie est basée sur les méthodes bien connues RS (Reed-Solomon) [102] et Viterbi [103]. Il a été décidé à la vue des performances attendues du système SACER de consacrer une taille identique à celle du PDU de niveau MAC (Figure II.11) pour la correction des erreurs (codage de type $\frac{1}{2}$).



Figure II.11 PDU de Niveau 2 et son champ de correction d'erreur.

Le Protocole de transmission est issu du standard ECMA 368 (Figure II.12) utilisant la technique « Orthogonal Frequency Division Multiplexing » (OFDM).

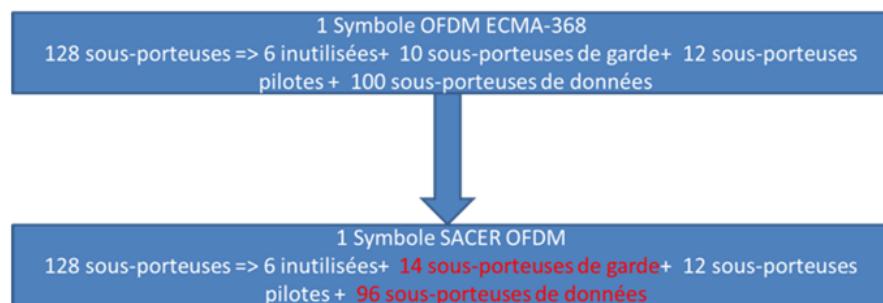


Figure II.12 Adaptation de la norme OFDM ECMA-368 au système OFDM SACER.

Chaque porteuse de données code 2 bits par Symbole OFDM. Ce qui signifie qu'un symbole OFDM SACER contient 192 bits de données, il faut donc émettre un second symbole pour transmettre les 192 bits de code correcteur d'erreur. Nous ne détaillerons pas plus le fonctionnement de ces couches protocolaires qui font l'objet de travaux de recherche à part entière. Le but ici étant de pouvoir être capable d'en créer un modèle d'abstraction.

II.2.5.2 Description de la couche réseau

La structure d'un réseau SACER est arborescente. Décrit précédemment, le réseau possède 3 classes d'équipement qui sont concentrateur, routeur et noeud. La conception du protocole réseau permet d'ajouter un élément de supervision mais qui n'entre pas dans le cadre de ces travaux. Dans le cas de SACER, un domaine de diffusion est composé d'un concentrateur, d'un routeur (ayant en réalité des fonctionnalités de Niveaux 1 et 2) et de 8 noeuds. Chaque concentrateur peut posséder jusqu'à 8 interfaces maximum qui se comportent comme des éléments de Niveau 0 dans la technologie MAC présentée précédemment. Un routeur est un élément de Niveau 1 qui organise un segment de Niveau 1 composé de 8 noeuds esclaves de Niveau 2.

Le schéma ci-dessous décrit la structure de la topologie : pour plus de clarté nous n'avons représenté que deux interfaces (au lieu de 8) d'un seul concentrateur, cas de notre étude par la suite.

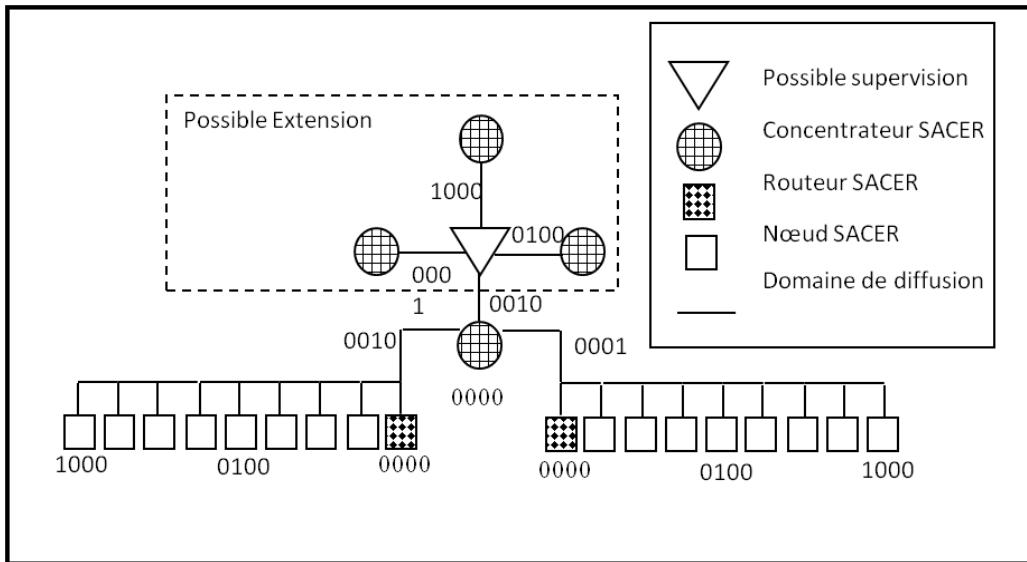


Figure II.13 Topologie logique du réseau SACER.

Les paquets réseau peuvent circuler selon un flux montant ou un flux descendant. Le format de paquet sera alors différent. Les PDU réseau qui seront encapsulés dans les trames MAC précédemment présentées sont alors les suivants (Figure II.14 et Figure II.15):



Figure II.14 PDU réseau descendant.



Figure II.15 PDU réseau ascendant.

Le **chemin réseau** est codé sur 12 bits. Les 4 premiers bits servent à identifier le chemin vers un concentrateur (dans le cas de l'extension présentant une supervision) c'est une adresse de réseau ; les 4 bits suivants identifient l'interface d'un concentrateur (segment de diffusion) : c'est une adresse de sous-réseau. Les 4 derniers bits identifient les éléments du segment de diffusion (routeur ou nœud). Le choix de 4 bits se justifie par le nombre d'interface/élément de segment, auquel s'ajoutent les identifiants de diffusion. Nous parlons de chemin par opposition à adresse en ce sens que la notion de source et de destination est implicite et dépend du flux de communication. Pour un paquet descendant le chemin identifie le(s) destinataire(s) du paquet. Dans le cas d'un paquet ascendant, si le chemin est un chemin de diffusion, il identifie également les destinataires, sinon il désigne l'émetteur du message.

Le champ **Protocole** permet le multiplexage de protocole de niveau applicatif. Sa valeur changera en fonction de l'application visée.

Les champs **Energie** et **SNR** présents dans les paquets ascendant permet de véhiculer des informations sur les nœuds terminaux à savoir : le niveau d'énergie dont ils disposent et le rapport signal sur bruit de leur liaison sans fil. En fonction de ces informations, des décisions pourront être prises par le système SACER.

II.2.5.3 Description de la couche Applicative

Les différentes applications du système SACER découlent naturellement du diagramme des cas d'utilisation de la Figure II.15. Nous retrouverons une application destinée à la remontée des données capteur et une application de configuration.

LE CAS NOMINAL : APPLICATION DE RELEVE DE MESURES

La première application auquel sera associé un numéro de protocole est relativement triviale, la figure suivante détaille son format de message. Une requête est envoyée dans le sens descendant, la réponse associée des noeuds suivra le même format. Ce cas d'utilisation correspond au cycle normal de communication que nous avons représenté dans la Figure II.6.



Figure II.16 Message applicatif de relevé de mesure.

Comme nous le verrons par la suite, la synchronisation est à la fois une spécification et un point dur du projet.

Le bit **Synchronisme** permet de préciser si le noeud considère que sa prise de mesure est synchrone avec le réseau, il est inutilisé dans le cas des messages descendants.

Le champ **N° de mesure** permet de préciser le numéro de mesure afin d'identifier un échantillon manquant. Cela assure la cohérence de la mesure vis-à-vis de la requête envoyée. La période d'échantillonnage des capteurs étant de $2^{15}=32768$ Hz, cela permet un recouvrement des numéros toutes les secondes.

Le champ **Validité** sur 8 bits indique dans le cas de la réponse quels sont les échantillons (les prises de mesures) considérés valides (pour lesquels un capteur fonctionnel a pu relever une mesure). Il est inutilisé dans le cas de la requête.

Les champs **Echantillon i** sont les valeurs échantillonnées par les capteurs du noeud. Ils sont évidemment inexploités par le message de requête.

LES CAS AUXILIAIRES: APPLICATION DE CONFIGURATION

La seconde application permet de traiter les autres cas d'utilisation de la Figure II.6. Le format de message adopté de cette application suit majoritairement un schéma de type requête-réponse. Le format de message générique est illustré dans la figure ci-dessous.



Figure II.17 Message applicatif de configuration.

Le Champ **Type de message** spécifie si le message est une requête ou une réponse, 14 autres cas non définis sont encore envisageables.

Le champ **Identifiant message** permet d'identifier le numéro de requête ou de réponse échangé.

Les champs suivants dépendront du type et de l'identifiant du message. Nous présenterons ici quelques cas de messages applicatifs à titre d'exemples.

La phase d'Initialisation ou la Découverte du réseau

Pendant cette phase, le concentrateur va diffuser un message **Hello_C** vers tous les routeurs. Chaque routeur à son tour, envoie un message **Hello_R** en broadcast vers tous les nœuds qui lui sont raccordés en précisant son adresse matérielle. Ces derniers vont répondre par un paquet **Rep_N**. Ensuite, les routeurs vont répondre par un message **Rep_R** au concentrateur. Cet échange entre les différents éléments du réseau est illustré par la Figure II.18.

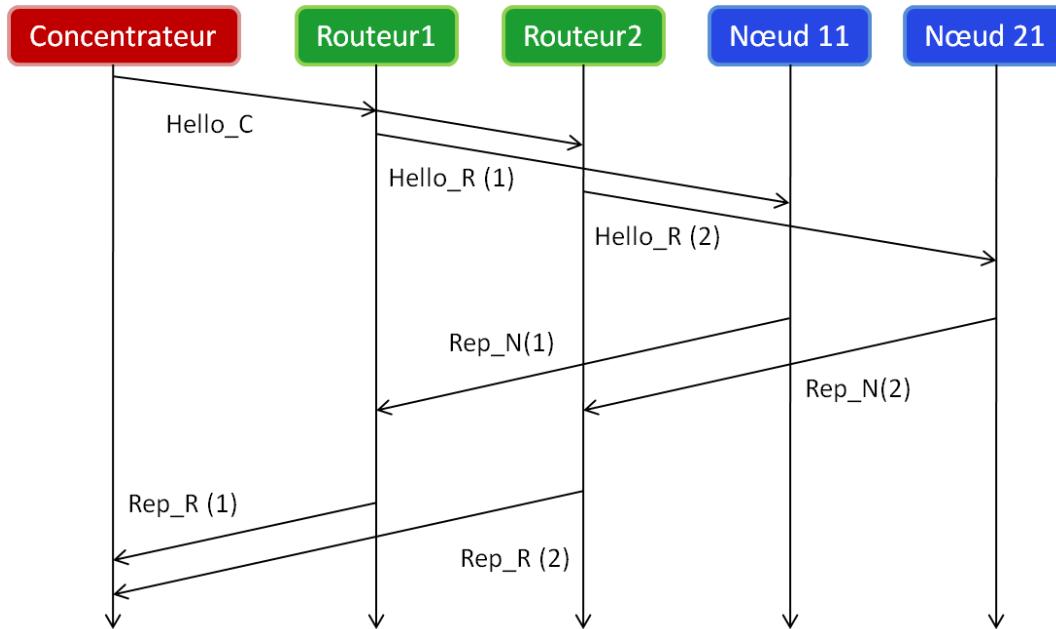


Figure II.18 Diagramme d'échange des messages pour la Découverte du réseau.

La Figure II.19 détaille le format d'un message de découverte du réseau :

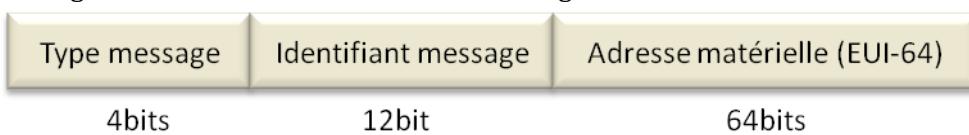


Figure II.19 Format d'un message applicatif de découverte.

La phase de configuration des capteurs

Ce cas d'utilisation est relatif au réglage des paramètres capteurs d'un ou plusieurs nœuds. Ces paramètres peuvent se matérialiser par le gain ou encore la valeur de la pleine échelle par exemple. Chaque paramètre d'un capteur peut-être consulté ou bien affecté, copié, sauvegardé, dupliqué etc. Les formats de message de la figure suivante décrivent le cas simple d'une lecture et d'une affectation.



Figure II.20 Format d'un message applicatif de réglage d'un paramètre capteur.

Le champ **Paramètre** identifie la caractéristique et le capteur concerné. Le champ **Accès**, lui, précise le type d'accès souhaité (lecture ou en écriture) dans le cas d'une requête,

ou le type du dernier accès effectué dans le cas d'une réponse. Le champ **Valeur** peut respectivement être la valeur à écrire, écrite ou lue pour les cas de : requêtes en écriture, réponse à une écriture ou enfin réponse à une lecture. Le champ **Valeur** est inutilisé pour une requête en lecture.

La phase de Bannissement

Le processus de bannissement ne s'effectue qu'au niveau du flux descendant du cycle de communication du réseau. Le message comportera, en plus des champs présents dans le message générique, les champs permettant d'identifier les nœuds bannis (Figure II.21).



Figure II.21 La trame de Bannissement.

Tout nœud recevant ce message vérifiera que l'adresse matérielle spécifiée lui correspond. Dans le cas où le message est envoyé à plusieurs nœuds, l'adresse matérielle devra correspondre à une adresse matérielle de diffusion.

La Figure II.22 montre le diagramme de séquence du bannissement dans le réseau :

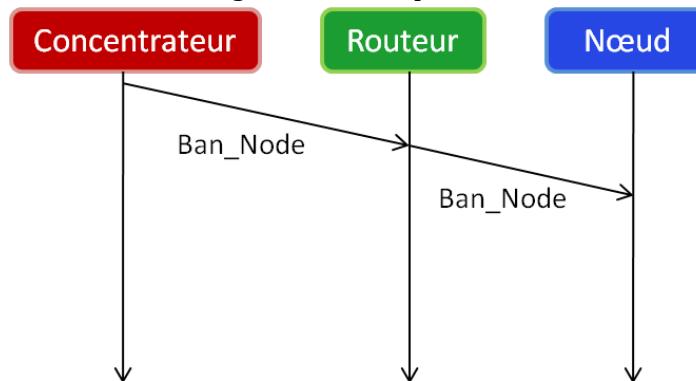


Figure II.22 Diagramme de séquence d'échange des trames du Bannissement.

La phase de synchronisation

Enfin, les messages applicatifs de type auxiliaires prévoient également des messages dédiés à la synchronisation. Le format et l'utilisation de ces messages n'est volontairement pas dévoilé dans ce chapitre, le protocole de synchronisation étant présenté en détails dans le chapitre suivant.

Pour résumer, nous présentons dans la figure suivante l'ensemble des trames pouvant circuler au sein du système SACER.

Trame SACER d'un flux descendant:



Trame SACER d'un flux ascendant :

**Message applicatif SACER en phase de mesure :****Message applicatif en phase de configuration :****Figure II.23 L'ensemble des trames pouvant circuler dans le système.**

II.3 Description de la plateforme matérielle

Le défaut est donc majeur : concevoir un produit final qui va s'adapter au mieux à l'environnement de déploiement, comme dans le cas d'un avion où l'ensemble des capteurs ne doivent en aucun cas modifier l'aérodynamisme des ailes et où il faut aussi veiller à ce que ces capteurs consomment un minimum d'énergie, d'où des contraintes de miniaturisation et d'autonomie.

Le matériel que nous avons utilisé a servi soit pour de tests (les cartes Wisair, les cartes IEEE-1588, etc.), soit pour du développement VHDL, matériel qui nécessite une phase de miniaturisation pour pouvoir être intégré dans le prototype.

Ainsi, par la suite nous présenterons l'environnement matériel utilisé durant et pour la conception des différents protocoles. Les résultats obtenus nous serviront par la suite dans la phase de modélisation en NS-2 de nos solutions d'amélioration du système.

II.3.1 Carte Wisair Wireless USB PCI Express Mini

Dans un premier temps, nous avons fait le choix des protocoles sur étagère UWB (l'UWB-OFDM et le MAC-WiMedia de cartes Wisair) qui semblent adaptés à nos besoins, mais qui se sont montrés inadéquats d'un point de vue performances pour la synchronisation, résultats qui seront présentés dans le dernier chapitre (chapitre V).

Les cartes Wisair (Figure II.24) ont été conçues pour offrir de très bonnes performances en terme de transmission sans fil. Elles sont basées sur une puce unique de type WSR601 et sur des technologies de couche PHY de type UWB-OFDM et MAC-WiMedia.

**Figure II.24 Carte Wisair Wireless USB PCI Express Mini.**

Un autre avantage de cette carte est de pouvoir coexister à proximité d'autres technologies sans fil, telles que Bluetooth et le 802.11. En ce qui concerne la gestion de l'énergie, l'utilisation de plusieurs modes de fonctionnement (ready, standby, sleep, and hibernation) minimise la consommation d'énergie.

Les principales caractéristiques de cette carte sont énumérées ci-dessous:

- ❖ Technologie Certified Wireless USB ; spécifications de la couche PHY et MAC de type WiMedia ;
- ❖ Débit de données jusqu'à 480 Mbit/s ;
- ❖ Bande de fréquences de 3,168 à 4,752 GHz (autorisée pour Europe) ;
- ❖ Consommation énergétique inférieure à 1.5W ;
- ❖ Antenne RF amovibles.

L'utilisation de ces cartes nous permet dans un premier temps de tester leurs performances, ainsi que le fonctionnement en présence d'autres types d'antennes (i.e. antennes directives).

Nous utiliserons également cette plateforme pour traiter un des points durs de ce projet, la synchronisation, par l'intégration et le test des cartes NI PCI-1588. Il faut noter que les campagnes de tests associées à la synchronisation nous ont permis de montrer une incompatibilité fonctionnelle entre les cartes Wisair et les cartes NI PCI-1588, qui se traduit par une dégradation de performances en terme de précision de la synchronisation. Par conséquent, les cartes Wisair n'ont pas été retenues dans le cadre de notre projet.

Nous présenterons en détails ces tests ainsi que les résultats obtenus dans le dernier chapitre de ce manuscrit (chapitre V).

II.3.2 Carte NI PCI-1588

Concernant toujours le problème de la synchronisation, nous avons décidé de tester le matériel utilisé aujourd'hui en filaire (cartes NI PCI-1588) pour une application sans fil.

Ainsi, ces cartes sont basées sur la norme IEEE-1588 Precision Time Protocol pour synchroniser les horloges des ordinateurs dans un réseau. La carte PCI-1588 utilise un connecteur standard RJ-45 Ethernet et fonctionne en mode maître-esclave. La fréquence et la phase de l'horloge présentes sur la carte sont ajustées automatiquement par un FPGA embarqué.

La carte PCI-1588 synchronise son horloge en générant des signaux synchronisés via plusieurs interfaces (Figure II.25):

- ❖ Des lignes PFI pour synchroniser des dispositifs autres que des PCI-1588 (comme des instruments traditionnels ou des automates via des connecteurs SMB) ;
- ❖ Une interface RTSI qui permet la synchronisation d'autres cartes PCI dans le même PC.

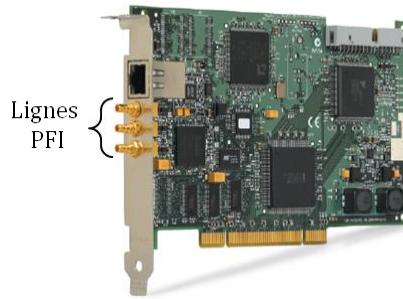


Figure II.25 Carte NI PCI-1588.

La grande précision d'horodatage fournie par le standard IEEE-1588, permet à ces cartes d'être utilisées pour corréler des évènements distribués.

En fonction de la connectique choisie par l'utilisateur, la déviation standard des horloges sera de :

- ❖ 33 ns pour une connexion directe de type Ethernet à travers un câble d'une longueur moyenne de 3m ;
- ❖ 48 ns pour une connexion via un hub ;
- ❖ 75 ns pour une connexion via un switch.

Nous pouvons donc constater que les performances de la norme IEEE-1588 correspondent aux contraintes de notre application. En contrepartie, les tests en présence d'une liaison sans fil se rapprochent de ces contraintes, sans pour autant les satisfaire. Cette norme va donc servir de point de départ pour l'élaboration d'une solution de synchronisation hybride.

Les résultats obtenus suite à ces campagnes de tests, ainsi que les résultats concernant la nouvelle solution de synchronisation seront présentés dans le chapitre V de ce manuscrit.

II.3.3 Plateforme ML410

Pour développer la partie réseau, nous avons choisi de travailler avec la carte ML410 de Xilinx. Cette carte à l'avantage de contenir un FPGA possédant à la fois une partie logique ainsi qu'une partie programmable en C.

Cette plateforme contient, principalement, les éléments suivants (Figure II.26):

- ❖ Un FPGA Virtex 4 XC4VFX60 ;
- ❖ Un processeur PowerPC FPGA's 405 ;
- ❖ Un processeur MicroBlaze FPGA's ;
- ❖ Des modules d'E/S qui s'étendent de l'UART à bas débit à la vitesse élevée du RocketIO Multi-Gigabit Transceivers (TMU) ;
- ❖ Une unité ATX d'alimentation ;
- ❖ Une mémoire de 64 Mo DDR ;
- ❖ Une mémoire amovible de 256 Mo DDR2 DIMM ;
- ❖ 4 Slots PCI de (3.3V et 5V) ;
- ❖ Un écran LCD pour l'affichage ;
- ❖ Un standard de communication JTAG pour la programmation du FPGA ;
- ❖ Une carte Compact Flash.



Figure II.26 La plateforme ML410 FPGA-Xilinx.

Cette plateforme est équipée d'un FPGA Virtex-4 FX conçu pour des applications embarquées demandant des hautes vitesses. A cet effet, plusieurs entrées/sorties hautes vitesses sont disponibles telles que les ports PCI express accessibles par l'intermédiaire des RocketIO Multi-Gigabit Transceivers (MGT, 622 Mb/s à 6,5 Gb/s).

Le tri-mode Ethernet MAC permet de bénéficier du Gigabit Ethernet sur 2 ports RJ45, mais aussi de vitesses plus conventionnelles : 10/100 Mb/s. Enfin, deux connecteurs d'extension à haute vitesse permettent de coupler la carte avec une autre sans diminuer les performances.

Au cœur du système, nous trouvons le Virtex-4 FX 60. Ce FPGA est idéal dans le cadre du travail qui doit être réalisé puisqu'il contient aussi une partie logique reprogrammable.

La partie logique peut contenir jusqu'à 200.000 cellules logiques, des tables LUT (Look-Up Table), des registres et des blocs RAM intégrés : 416 kB. La conception peut être faite soit en VHDL, soit en Verilog par l'intermédiaire des logiciels fournis par Xilinx.

Ils peuvent aussi être équipés de deux blocs mémoires : l'un de 64 Mo DDR intégré à la carte et l'autre de 256 Mo DDR2 (à connecter).

Enfin, le XC4VFX60 comporte 128 blocs XtremeDSP (Digital Signal Processing), chaque bloc est constitué de deux blocs DSP48. Ces derniers sont équipés d'additionneurs de multiplicateurs, d'accumulateurs qui combinés ensemble, permettent de créer des fonctions complexes sans avoir besoin de ressources logiques. Il est par exemple possible de mettre en place pour le traitement du signal différents types de filtre FIR (Filtre à réponse Impulsionnelle Finie).

L'intérêt d'utiliser ces types de cartes est qu'elles permettront le développement des couches basses par les électroniciens, ainsi que le développement de nos propositions concernant les couches hautes, sur le même support matériel.

II.4 Modélisation sur NS-2

Pour la phase de simulation, le choix du logiciel (NS-2) a été justifié dans le premier chapitre. Son caractère open source nous a permis de définir notre propre protocole de

routage, et d'apporter les modifications nécessaires sur les autres couches de communication.

Avant de faire un rappel des algorithmes que nous avons implémentés en NS-2 (directement liés à notre application), nous considérons utile de donner quelques lignes directives sur la façon générale de développer de nouveaux objets en NS-2.

II.4.1 Eléments de base de la simulation en NS-2

Les unités fondamentales d'échange entre les objets de simulation NS-2 sont les objets de la classe `Packet`. Un paquet est généralement composé d'un en-tête et de ses données utiles, qui contiennent les informations nécessaires pour sa « livraison ».

➤ *Un nouveau paquet :*

La création d'un nouveau protocole de communication nécessite, en fonction de sa complexité, la définition des nouveaux en-têtes de paquet, l'utilisation des en-têtes déjà existants, ou d'une simple extension de ceux existants.

L'avantage de l'en-tête créé spécialement pour un nouveau protocole, est qu'il va stocker seulement les attributs utilisés par le protocole en question (tels que l'ID unique, la taille de paquet, le type de paquet, etc.). L'en-tête spécifique du paquet est représenté en NS-2 par une structure de données de type « `struct` » (où sont stockés les attributs avec leurs valeurs), qui fournit également une fonction d'accès (`p`). Cette fonction retourne la référence de l'en-tête du paquet au protocole concerné.

Dans [104], nous pouvons trouver un guide permettant de modifier les fichiers sources de NS-2, pour la création d'un nouveau protocole.

➤ *Un nouvel agent :*

L'agent est un objet NS-2, qui est responsable de la création et de la destruction des paquets. Il existe deux catégories d'agents: les agents de niveau routage et les agents de la couche transport. Le premier a pour rôle d'envoyer et de recevoir des paquets de contrôle, donc de commander le protocole de routage. L'agent de la couche transport contrôle la congestion et la fiabilité d'un flux de données reçues à partir d'une application associée.

En résumé, un agent fait la liaison entre l'application et les couches de bas niveau (MAC, PHY). Dans la section 10 de [105], nous pouvons identifier les principales étapes dans la création d'un nouvel agent:

- ❖ Décider de la structure d'héritage et définir les classes appropriées ;
- ❖ Définir les méthodes « `recv()` » et « `timeout()` » ;
- ❖ Définir les autres classes nécessaires ;
- ❖ Définir les fonctions de liaison OTcl et écrire le code associé qui permet l'accès à l'agent.

➤ *Une nouvelle application :*

Une application fonctionne au dessus de la couche transport avec ses agents associés, et modélise les besoins utilisateur pour la transmission de données. Il existe deux principaux types d'applications: les générateurs de trafic et les applications simulées. Les générateurs de trafic sont basés sur un calendrier prédéfini lors de la création de flux utile

de données. Une application simulée va créer la demande si elle est uniquement en cours d'exécution.

Un agent et une application peuvent être connectés en utilisant la commande de classe Application, dont la syntaxe est la suivante:

\$app attach-agent \$agent (II.1)

où \$app et \$agent sont des objets de classe Application, respectivement Agent.

Grâce aux pointeurs agent_ et app_, les nouveaux objets ainsi connectés peuvent appeler les fonctions publiques de leurs classes. Les plus importantes fonctions utilisées sont : « send (nbytes) » et « recv (nbytes) » de la classe Application, et « sendmsg (nbytes) » de la classe Agent.

La classe Application fournit également les fonctions protégées « start () » et « stop () » pour démarrer et arrêter un objet de type Application. Le lecteur pourra trouver plus de détails dans [105].

Pour avoir une image plus complète de l'interface de programmation (Application Programming Interface ou l'API) d'un nœud mobile en NS-2, nous présentons dans la Figure II.1 la pile protocolaire associée.

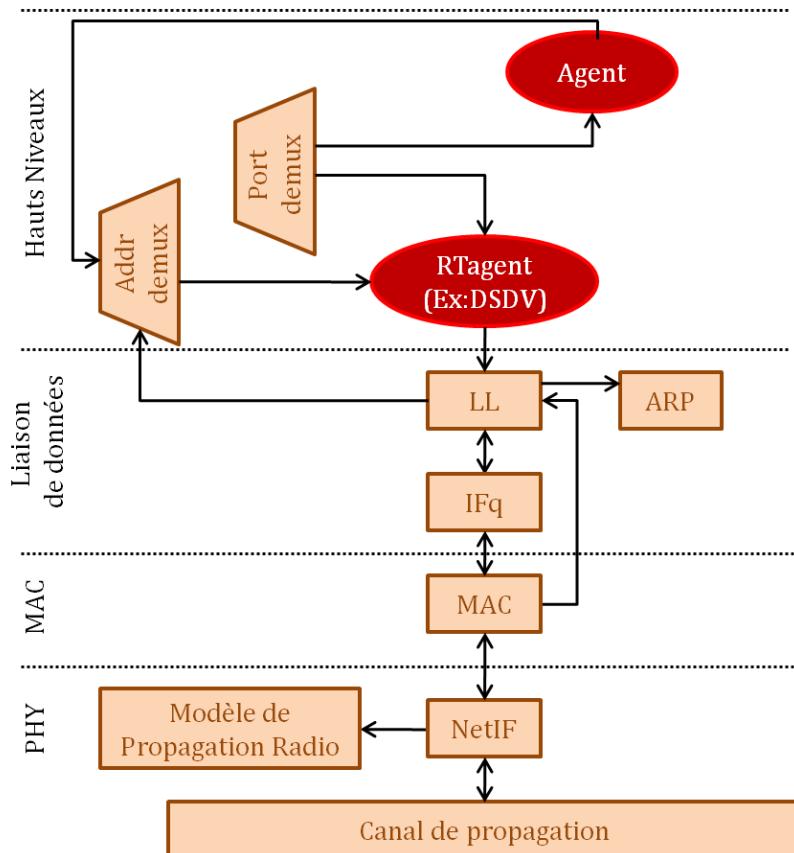


Figure II.1 Architecture d'un nœud mobile implémenté en NS-2.

Nous pouvons identifier les blocs ainsi présentés avec la pile protocolaire du modèle ISO/OSI comme suit :

- ❖ Le niveau physique : s'identifie dans une interface réseau (NetIF) avec des antennes omnidirectionnelles, il utilise un modèle de propagation radio et qui assure l'interconnexion de ces composants avec le canal de propagation ;

- ❖ Le niveau MAC : nous pouvons choisir entre plusieurs types de couche MAC (802.11, 802.15.4, TDMA, etc.) ou redéfinir son propre protocole à partir de ceux-ci ;
- ❖ Le niveau liaison de données : intègre une interface de file d'attente (IFq) avec un protocole ARP (Address Resolution Protocol) ;
- ❖ Le niveau réseau : correspond à des entités de type RAgent qui sont des algorithmes de routage comme TORA (Temporally Ordered Routing Algorithm), AODV (Ad-hoc On-demand Distance), DSDV (Destination Sequenced Distance-Vector), etc. ;
- ❖ Le niveau transport : correspond aux entités Agent, qui sont directement liées au type de l'application ;
- ❖ Le niveau applicatif : s'identifie avec des agents spécifiques, mentionnés ci-dessus. Cette description de la création de nouveaux objets de simulation sert de guide pour comprendre les changements que nous avons fait aux fichiers sources de NS-2, dans le but de simuler les protocoles et les applications souhaités.

II.4.2 Les principales contributions en NS-2

Précédemment mentionnée, la topologie du réseau est fixe (les émetteurs ne se déplacent pas les uns par rapport aux autres). Il y a un point central unique (le concentrateur) qui communique avec des points intermédiaires (les routeurs), qui eux même communiquent avec les éléments terminaux (les noeuds).

Pour réaliser cette architecture hiérarchique dans la simulation, nous avons défini un protocole spécifique au niveau de la couche transport de NS-2 et aussi de modifications au niveau de la couche MAC. Au niveau de la couche réseau, nous avons choisi d'utiliser les principes de NOAH (No ad-Hoc Routing Agent), qui est un protocole de routage qui nous permettra de spécifier notre propre routage (i.e. celui défini dans le projet SACER).

Le canal de communication est de type sans fil ; il simule la communication à travers un modèle de propagation Tow Ray Ground avec des antennes omnidirectionnelles. La couche liaison de données choisie est de type MAC_TDMA.

Ainsi, au lieu d'utiliser les antennes existantes sur NS-2 (omnidirectionnelles), nous allons implémenter et tester les avantages des antennes directives. Une autre amélioration envisagée concerne l'utilisation des interfaces multiples pour certains éléments du réseau.

Un aspect qui va aussi être abordé concerne la synchronisation des données, qui, dans notre application, a une importance majeure.

Nous rajoutons que l'architecture de communication imposée par NS-2 implique un certain nombre d'approximations, comme :

- ❖ Une couche Wireless-PHY et MAC gérés par paquet,
- ❖ Un adressage IP imposé, etc.

Dans ce contexte, nous avons implémenté les cas d'utilisations que nous avons présentés auparavant.

Dans la section suivante, nous avons l'intention de présenter l'implémentation de la topologie réseau de notre système de communication. Toutes les autres contributions avec leurs résultats associés vont être discutées en détails dans les chapitres III (Proposition d'un nouveau protocole de synchronisation), IV (Les antennes directives dans les réseaux des capteurs sans fil) et V (Simulations et expérimentations).

II.4.2.1 *Implémentation de notre topologie réseau*

Commençons par rappeler que l'organisation de notre réseau est hiérarchique et qu'elle se compose d'un concentrateur, huit routeurs et huit nœuds pour chaque routeur, par aile. Cette organisation est certes héritée du projet SACER, mais nous avons souhaité envisager une approche plus générale sur l'architecture réseau des composants qui sera explicitée dans le paragraphe suivant. Pour implémenter cette stratégie, nous avons réalisé un certain nombre de restrictions de communication. Plus exactement, nous avons créé une commande Tcl nommée « alloc_ip », qui a le rôle « d'associer » les éléments du réseau censés communiquer ensemble. Ainsi, à l'aide de cette fonction et d'un certain nombre de modifications au niveau de la couche MAC, nous réalisons un filtrage du trafic, par adresse IP, ce qui nous permet d'obtenir le trafic de données utiles, présenté dans la Figure II.2.

La session de communication commencera au niveau du **concentrateur** par la création au niveau de la couche applicative d'un paquet avec l'identifiant «pkt_no = 150». Ce paquet va être transféré au niveau transport (à l'agent) et diffusé vers tous les routeurs. Après la réalisation d'un cycle complet, le concentrateur recevra de la part de chaque routeur un message avec l'identifiant «pkt_no = 400». Il faut noter que les valeurs choisies pour l'identifiant paquet n'ont pas d'influence sur les résultats de simulation.

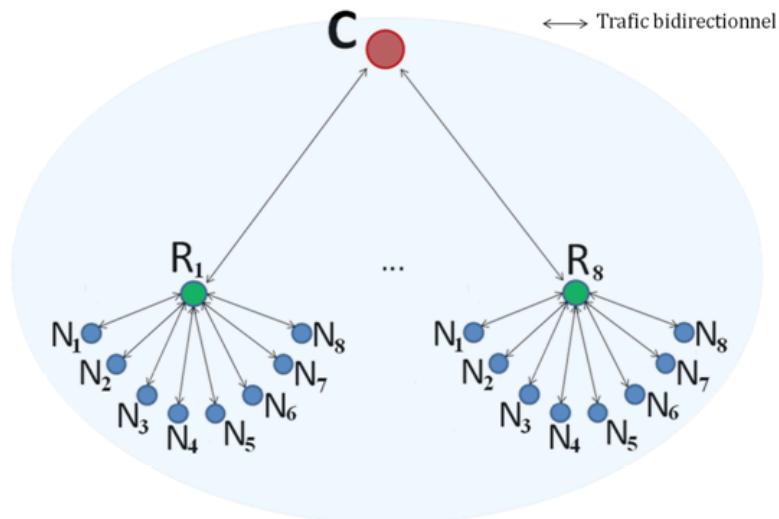


Figure II.2 Trafic de données utiles dans notre réseau hiérarchique.

Concernant les **routeurs**, chacun reçoit de la part du concentrateur le message avec l'identifiant «pkt_no = 150» et diffuse vers leurs nœuds associés un message avec l'identifiant «pkt_no = 200». Lorsque **les nœuds** reçoivent de la part du routeur le message de données avec l'identifiant «pkt_no = 200», chacun d'eux va créer une réponse avec l'identifiant «pkt_no = 300», qui va être envoyée directement au routeur associé. Après avoir reçu les réponses de l'ensemble des huit nœuds, le routeur enverra un message avec l'identifiant «pkt_no = 400» directement au concentrateur.

Ce n'est qu'après la réception de tous ces messages que le concentrateur redémarrera une nouvelle session de communication.

Ainsi, à travers cette configuration du réseau, nous sommes en mesure d'assurer la connectivité de tous les éléments du réseau et d'évaluer la puissance de transmission nécessaire à l'émission de chaque message.

II.4.2.2 Proposition d'utilisation d'interfaces multiples

Comme vu précédemment, le routeur est un élément indispensable dans notre système pour assurer une bonne connectivité entre les éléments du réseau ainsi qu'une bonne conservation de l'énergie (les distances entre les objets communicants sont plus petites).

En même temps, il ne faut pas oublier qu'une tâche difficile dans un réseau sans fil reste le contrôle de congestion. Notre solution à ce problème consiste à utiliser un protocole d'accès au médium de type TDMA. Cette technique permet à plusieurs utilisateurs de partager la même fréquence sans provoquer d'interférence, parce que le signal est divisé en multiples intervalles de temps. Étant donné que le nombre d'éléments dans notre système est élevé (2*73 éléments), l'utilisation de cette technologie peut conduire à des retards de transmission importants.

Ainsi, une solution possible à ce problème, qui permet également de préserver les avantages du TDMA, est d'utiliser des interfaces multiples, c'est-à-dire d'avoir deux ou plusieurs interfaces de communication par élément.

Nous avons alors 2 possibilités pour placer ces interfaces multiples :

- ❖ Soit au niveau du routeur ;
- ❖ Soit au niveau du concentrateur.

Il n'est pas nécessaire de le déployer sur les 2 niveaux en même temps, un seul suffit pour éviter les interférences.

La première possibilité est donc de les placer au niveau du routeur. Dans ces conditions, notre système sera composé de plusieurs sous-réseaux qui communiqueront entre eux à travers les routeurs, représentés dans la figure suivante (Figure II.3):

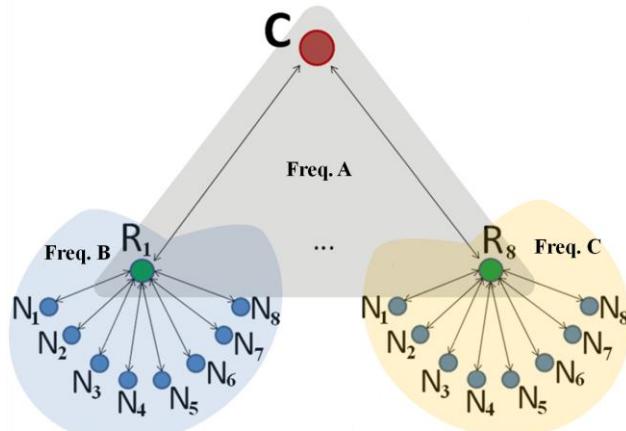


Figure II.3 Organisation du réseau avec des interfaces multiples (Solution I).

Malheureusement, cette nouvelle architecture implique également un inconvénient: le routeur R₁ qui communique sur deux fréquences (fréquences A et B) aura une complexité beaucoup plus élevée par rapport à un nœud capteur, ce qui va augmenter le coût total de fabrication (qui est très important, surtout pour les applications industrielles). Donc, pour maintenir un coût de production raisonnable, le processus de fabrication d'un routeur doit être similaire à celui d'un nœud capteur.

La solution alternative consiste alors à adopter une configuration du système avec plusieurs sous-réseaux, mais dans laquelle seulement le concentrateur est conçu selon la méthode des interfaces multiples (Figure II.4) :

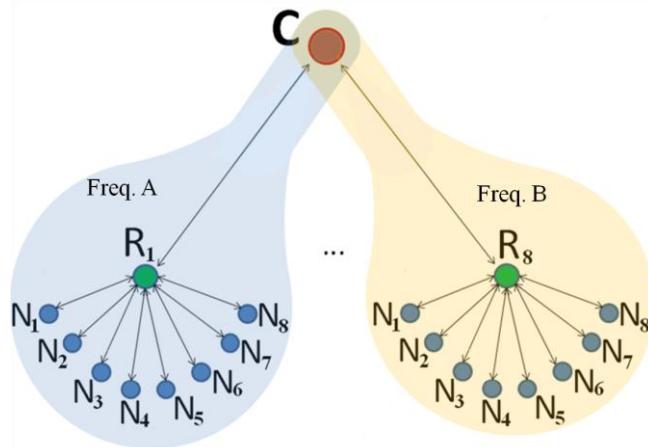


Figure II.4 Organisation du réseau avec des interfaces multiples (Solution II).

Cela implique une architecture composée de huit interfaces de communication pour le concentrateur, ce qui signifie que nous aurons huit sous-réseaux interconnectés. De plus, comme le concentrateur aura été imaginé comme un élément plus complexe que les autres, l'introduction de plusieurs interfaces n'a pas d'impact négatif sur la faisabilité du système. Au contraire, nous allons démontrer à travers des simulations (chapitre V) que cette nouvelle disposition présente plusieurs avantages pour l'ensemble de notre système.

II.5 Conclusion

Nous avons présenté lors de ce chapitre le contexte dans lequel s'est déroulé notre travail, ainsi qu'une description de l'application du point de vue des besoins, des contraintes et des propositions, en tenant compte du cahier des charges.

Plus exactement, nous avons abordé deux premières phases dans la réalisation d'une application, à savoir :

- **La phase d'analyse** - qui consiste à étudier le système, à identifier les acteurs intervenants dans ce réseau, à dégager les cas d'utilisation, à travers l'étude des besoins et des contraintes fonctionnelles. Cette étape vise à mieux saisir le fonctionnement global du réseau afin de faciliter la concrétisation de l'étape suivante ;
- **La phase de conception** – qui consiste à présenter la méthodologie de conception adoptée, ainsi qu'à détailler les différents modules caractérisant notre application. Lors de cette étape, nous avons défini le fonctionnement des différents modules à développer et ce, au moyen d'organigrammes fonctionnels.

Nous avons ensuite proposé l'organisation de notre système : une topologie hiérarchique, avec une couche MAC de type TDMA. A cette configuration, nous avons apporté une amélioration basée sur le principe de communication avec interfaces multiples.

L'implémentation en NS-2 de ces propositions va nous permettre d'étudier leur comportement. Ainsi, nous passerons dans les chapitres qui suivent à la validation par simulation des différents modules proposés, ce qui va permettre l'implémentation des protocoles ainsi développés sur un FPGA.

III PROPOSITION D'UN NOUVEAU PROTOCOLE DE SYNCHRONISATION

| | | |
|--------------|--|------------|
| III.1 | Introduction | 83 |
| III.2 | Problématique de la synchronisation | 84 |
| III.2.1 | Stratégies possibles | 84 |
| III.2.2 | Analyse | 85 |
| III.2.3 | Bilan | 85 |
| III.2.4 | Le standard IEEE-1588 | 86 |
| III.2.5 | Le protocole de synchronisation PBS | 88 |
| III.3 | Présentation de la contribution: l'IEEE1588-PBS hybride | 90 |
| III.3.1 | Algorithme et fonctionnement | 91 |
| III.3.2 | Intégration dans notre cas d'étude | 92 |
| III.3.3 | L'impact sur la consommation d'énergie | 94 |
| III.4 | Implémentation et mise en œuvre | 94 |
| III.4.1 | Conditions pratiques de mise en œuvre | 94 |
| III.4.2 | Les étapes d'implémentation dans le simulateur NS-2 | 95 |
| III.5 | Conclusion | 100 |

Nous proposons dans ce chapitre de développer un nouveau protocole qui aura pour but de garantir une bonne précision de la synchronisation. L'état de l'art des solutions existantes, présenté au premier chapitre, nous a montré que les performances pour la précision de la synchronisation étaient insuffisantes et nous guide à tenir compte des besoins de minimisation de la consommation d'énergie dans le contexte de notre application. Pour cela, nous allons décrire en détails le fonctionnement d'une nouvelle solution, dont l'analyse des résultats sera présentée dans le Chapitre V.

III.1 Introduction

Nous avons pu constater que le domaine des réseaux de capteurs sans fil (WSNs) est aujourd'hui intensivement analysé et étudié, la principale motivation étant leur grande complexité.

L'un des problèmes les plus importants est lié à leur consommation d'énergie, car cela a un impact majeur sur la durée de vie du réseau. Une autre demande pour de nombreuses applications est d'assurer la datation des données transmises, ce qui conduit à une consommation d'énergie supplémentaire (des messages supplémentaires vont être reçus à chaque nœud).

La durée de vie d'un réseau devient ainsi un compromis entre la qualité des services, car investir plus d'énergie peut augmenter la qualité, et inversement d'un point de vue durée de vie.

Pour cela, l'idée d'harmoniser ces deux compromis est née, et servira comme principe de base dans notre démarche de conception d'un nouveau protocole de synchronisation pour satisfaire les besoins de précision.

III.2 Problématique de la synchronisation

III.2.1 Stratégies possibles

Les spécifications fonctionnelles discutées dans le chapitre précédent ont fait remonter des besoins de prise de mesures quasi-synchrones, demande essentielle pour la validation des modèles de simulation. Plus exactement, le décalage toléré entre deux prises de mesures sur deux capteurs différents du système doit être, pour les cas les plus contraints, inférieur à 500 ns. Il en résulte des contraintes de datation comparables, ce qui nous oblige à considérer dans la conception du système, l'utilisation d'horloges réparties synchrones avec une précision de l'ordre de 500 ns.

La mise en œuvre de ces horloges synchrones est particulièrement complexe car les tolérances imposées se situent à la limite de ce qu'offre aujourd'hui la littérature dans le contexte des réseaux d'instrumentation filaires. Avant de se lancer dans la conception, nous avons évalué les solutions classiques de synchronisation, susceptibles d'être compatibles avec notre application.

LE GPS

Le système GPS possède un mécanisme de synchronisation d'horloge interne du récepteur avec le temps fourni par les satellites. Les satellites GPS envoient donc des signaux qui sont pilotés par une horloge atomique d'une très grande précision. Les récepteurs possèdent généralement une horloge à base de quartz qui est moins stable, et les mesures de signaux corrigent les dérives du temps. Des mesures effectuées avec des récepteurs situés au sol à grande distance, montrent qu'ils se synchronisent entre eux avec une précision de l'ordre de quelques nanosecondes [106].

Ainsi, même si les performances des solutions basées sur l'utilisation du signal GPS correspondent à nos besoins, elles ont été écartées pour des raisons inhérentes à l'application avionique (les conditions météo variable, les réflexions etc.), et en particulier liées à la difficulté de placer des capteurs sous l'aile de l'avion.

LES SOLUTIONS AD-HOC

Une autre alternative serait de faire appel à des solutions ad-hoc qui utilisent une synchronisation externe sur un canal séparé [107]. Cette possibilité a été également écartée pour des raisons de consommation et de complexité des chaînes de réception.

LES PROTOCOLES STANDARDS/RESEAU

Nous nous fixons donc comme objectif, d'utiliser un standard de synchronisation issu des réseaux d'instrumentation filaires : IEEE-1588. Ce standard offre des performances de synchronisation de l'ordre de 50 ns sur des réseaux Ethernet dédiés, mais il n'est pas initialement conçu pour des réseaux sans fil.

En conséquence, cet aspect nous a mené à réaliser des tests sur les cartes de transmission UWB WiMedia (Wisair Wireless USB PCI Express Mini) décrites dans le chapitre précédent. Lors de ces tests, il a été mesuré une précision pour la synchronisation de l'ordre de 200 μ s, ce qui est très supérieur à nos contraintes.

Une étude complémentaire a donc été lancée pour évaluer l'impact d'une liaison sans fil sur le fonctionnement du protocole IEEE-1588. La littérature étant peu détaillée sur le sujet [108], [109], [110], nous avons réalisé des expérimentations permettant d'évaluer la raison du mauvais fonctionnement du protocole IEEE-1588 sur les cartes USB WiMedia. Cette campagne de mesures est présentée en détails dans le dernier chapitre de ce mémoire (Simulations et expérimentations).

Afin de répondre au besoin de la synchronisation de notre application, nous allons nous intéresser dans la suite à l'évaluation de l'IEEE-1588 en présence d'une connexion sans fil.

III.2.2 Analyse

Nous notons tout d'abord que l'évaluation de l'IEEE-1588 en présence des cartes UWB WiMedia a eu des résultats pour la précision de la synchronisation ($\sim 200 \mu$ s) qui ne sont pas satisfaisantes pour notre application. Plus précisément, cet aspect nous a conduit à la conclusion qu'il va falloir proposer une solution qui sera capable de répondre aux besoins de notre application (en termes de précision de synchronisation et d'économie d'énergie).

Ainsi, les mesures effectuées montrent l'influence de l'asymétrie du lien de communication sur la précision de la synchronisation offerte par le protocole IEEE-1588. Nous avons obtenu une asymétrie de délai de 6 ms entre la voie aller et la voie retour sur les cartes Wisair Wireless USB PCI Express Mini, alors qu'elle est d'environ 100 μ s sur une liaison Ethernet. Cette asymétrie s'explique par le fonctionnement du protocole MAC de la carte Wisair Wireless USB PCI Express Mini qui implémente le paradigme maître-esclave, où le maître a une plus forte priorité d'accès au canal.

Dans ce contexte, nous rappelons que le protocole IEEE-1588 fait l'hypothèse d'un lien symétrique dans la description de son fonctionnement [111].

En ce qui concerne l'influence de la gigue dans la comparaison UWB WiMedia/Ethernet, nous constatons que ce n'est pas le problème principal. Cette conclusion vient du fait que les mesures de gigue sont comparables pour les deux cas (autour de la centaine de μ s).

Nous proposons ainsi d'utiliser une couche MAC capable de garantir ces contraintes en terme de limitation de la gigue (max 100 μ s) et de symétrie de lien afin de garantir le bon fonctionnement du protocole de synchronisation. Ces valeurs seront prises en compte dans la conception des couches MAC et Réseau.

III.2.3 Bilan

Après avoir étudié les différentes solutions de synchronisation existantes dans la littérature (chapitre I) et après avoir discuté les performances de l'IEEE-1588 à travers une liaison sans fil, nous avons l'intention par la suite de proposer un nouveau protocole de

synchronisation qui doit être capable de répondre aux principaux besoins de notre application : une précision de la synchronisation inférieure à 500 ns ainsi qu'une capacité d'économiser l'énergie.

L'étude des principales solutions et protocoles déjà existants nous a guidés dans la définition de la nouvelle proposition. Ainsi, le Tableau III.1 présente une comparaison entre ces solutions en prenant en compte un certain nombre de paramètres, où N représente le nombre de paquets échangés dans un cycle de synchronisation et L le nombre de nœuds présents dans le réseau.

En analysant le Tableau III.1, nous pouvons constater que les protocoles qui ont une très bonne précision de synchronisation présentent une forte consommation en énergie, ou vice versa.

Notre proposition sera donc de mettre en œuvre un protocole de synchronisation basé sur le standard IEEE-1588 qui est très performant mais qui a été conçu pour des réseaux locaux filaires, et le protocole PBS, dédié aux réseaux de capteurs, mais qui n'est pas en mesure de fournir la précision de la synchronisation nécessaire pour notre application.

| Protocole | Paramètres de performance | | | | |
|----------------|---------------------------|-------------------------|---------------------|------------------------|-------------------------|
| | Schéma utilisé | Nombre messages/ cycles | Précision d'horloge | Couche | Consommation en énergie |
| IEEE-1588 [52] | Saut unique | $4*N*L$ | 50-200 ns | Application / Physique | Elevée |
| RBS [54] | Récepteur-Récepteur | $N*L2$ | 29.1 μ s | Application | Elevée |
| TPSN [55] | Emetteur-Récepteur | $2*N*L$ | 16.9 μ s | MAC | Moyenne |
| FTSP [56] | Emetteur-Récepteur | $N*L$ | 1.7 μ s | MAC | Faible |
| PBS [57] | Multi sauts/ saut unique | $2*N$ | 29.1 μ s | Application | Faible |

Tableau III.1 Classification des protocoles de synchronisation.

Les principaux objectifs de cette solution hybride sont les suivants: assurer une bonne précision (centaines de ns) pour les horloges des éléments du réseau et fournir une économie d'énergie importante pour le réseau.

Afin de réaliser une présentation de la contribution, nous proposons dans la suite de détailler les principes de fonctionnement des deux protocoles de synchronisation qui seront à la base de notre solution hybride.

III.2.4 Le standard IEEE-1588

Le protocole IEEE-1588 est un protocole standardisé qui permet d'obtenir une synchronisation entre les périphériques d'un réseau avec une précision inférieure à la microseconde. Une telle performance a l'avantage de permettre des extensions plus sophistiquées pour les applications envisagées.

Son principe consiste à synchroniser les horloges esclaves avec l'horloge d'un maître, en veillant à ce que les événements et les horodatages de tous les nœuds utilisent la même base de temps.

La correction de la déviation des horloges comporte deux phases (la correction de décalage (l'offset) et la correction de retard (le délai)), parce que la différence de temps entre l'horloge maître et les horloges esclave est une combinaison du décalage d'horloge et de retard de transmission du message.

Dans la première phase de synchronisation, le nœud maître commence la correction de l'offset en envoyant deux messages : le sync et le follow_up. Lorsque l'esclave reçoit le message de synchronisation envoyé par le maître, il utilise son horloge locale pour l'estampiller avec le moment d'arrivée (Figure III.1).

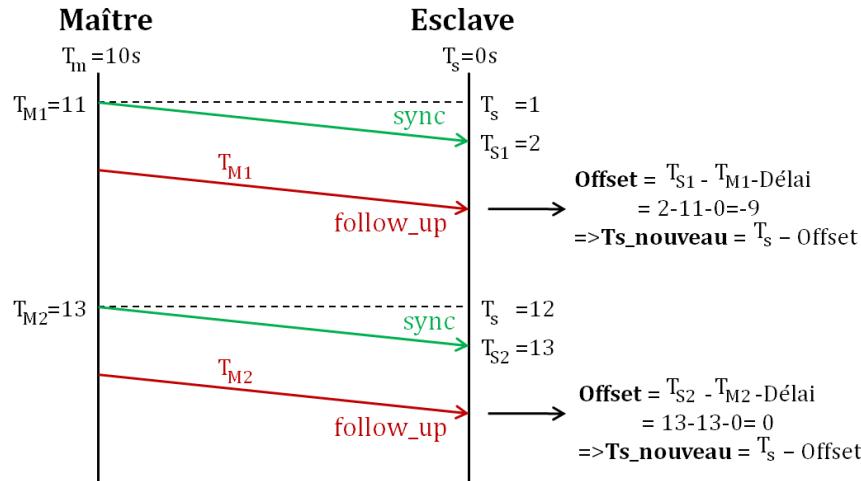


Figure III.1 La première phase de synchronisation de l'IEEE-1588.

Après avoir reçu le message follow_up, l'esclave va comparer son heure locale avec l'estampillage de transmission du message sync, effectué par le maître. La différence entre les deux horodatages représente la différence entre les deux horloges internes, à laquelle se rajoute le retard de propagation du message.

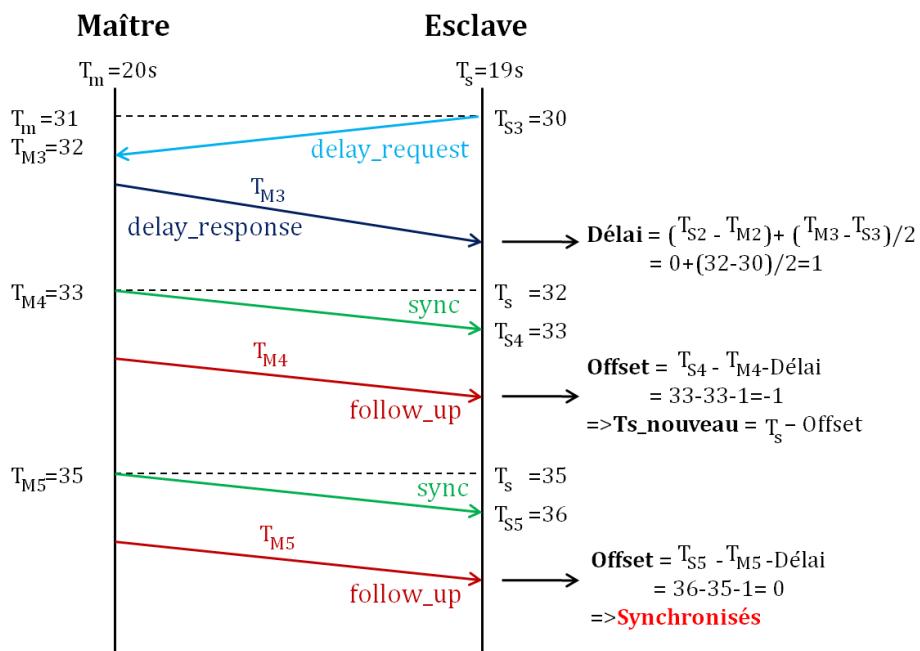


Figure III.2 La deuxième phase de synchronisation de l'IEEE-1588.

La deuxième phase illustrée dans la Figure III.2 a comme but de déterminer le délai de propagation entre le maître et l'esclave.

Ainsi, les deux nœuds vont horodater l'envoi, et l'arrivée du message `delay_request`. Le maître répondra à l'esclave avec un message `delay_response`, qui contient l'heure d'arrivée du message `delay_request`. A ce moment, l'esclave a toutes les informations pour calculer le retard de propagation.

Après avoir calculé le délai de propagation, l'esclave ajuste son horloge pour être synchrone avec l'horloge du maître. Parce que les horloges internes (que ça soit du maître ou des esclaves) ne sont pas parfaites et dérivent d'une manière indépendante, des cycles périodiques de synchronisation sont nécessaires pour maintenir cette synchronisation.

En résumé, le décalage et le délai de propagation entre l'horloge interne de l'esclave et celle du maître, sont calculés comme suit:

$$\text{delay} = \frac{(T_{S2} - T_{M2}) - (T_{M3} - T_{S3})}{2} \quad (\text{III.1})$$

$$\text{offset} = T_{S1} - T_{M1} - \text{delay}$$

Après avoir décrit le mode de fonctionnement du standard IEEE-1588, nous considérons maintenant pertinent de revenir sur un travail mentionné dans la partie I.3.2.2 de notre état de l'art. Ainsi, les auteurs de [53] ont proposé une mise en œuvre de l'IEEE-1588 sur un nouveau type de nœud capteur. Ce que nous retenons c'est que le processeur utilise un oscillateur à quartz compensé en température (TCXO), afin de réduire le taux de sa dérive. Il a une tolérance en fréquence de 1,5 ppm (parts per million), qui se traduit par une dérive maximale de 1,5µs par seconde.

Leurs expérimentations ont abouti à une précision de la synchronisation entre l'esclave et le maître d'approximativement 200 nanosecondes, ce qui va servir comme référence pour notre future implémentation.

III.2.5 Le protocole de synchronisation PBS

Le protocole PBS [57] est une approche pour la synchronisation des réseaux à grande l'échelle, qui est parfois aussi nommée ROS (Receiver-Only Synchronization). Cette appellation est directement liée à son algorithme de fonctionnement (Figure III.3).

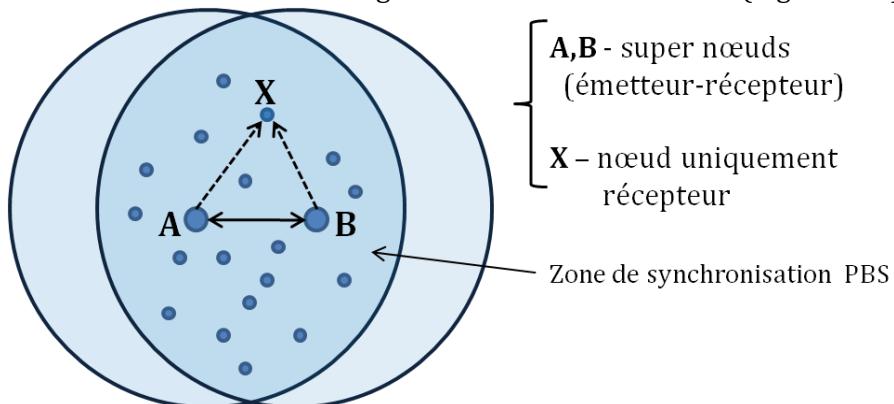


Figure III.3 Principe de synchronisation du protocole PBS.

Chaque nœud qui se trouve dans la zone de couverture de la paire A-B (par exemple le nœud X), recevra les messages diffusés par les deux nœuds (A et B).

Supposons maintenant que le nœud A et le nœud B essaient de se synchroniser à travers un échange bilatéral de messages de synchronisation, où B est le nœud parent. Le nœud X recevra une série de messages de synchronisation contenant les estampillages de la paire A-B. Par conséquent, en utilisant les informations d'horodatage, le nœud X peut se synchroniser avec le nœud parent B, sans envoyer aucun message.

Les nœuds A et B peuvent être considérés comme des « super nœuds », car ils fournissent les informations de synchronisation pour tous les nœuds situés à leur portée.

L'échange de messages est ainsi représenté dans la Figure III.4, où $\theta_{\text{offset}}^{(PA)}$ et $\theta_{\text{offset}}^{(PB)}$ représente le décalage (ou l'offset) des horloges entre le nœud A et le nœud B, et où les messages de synchronisation sont échangés N fois.

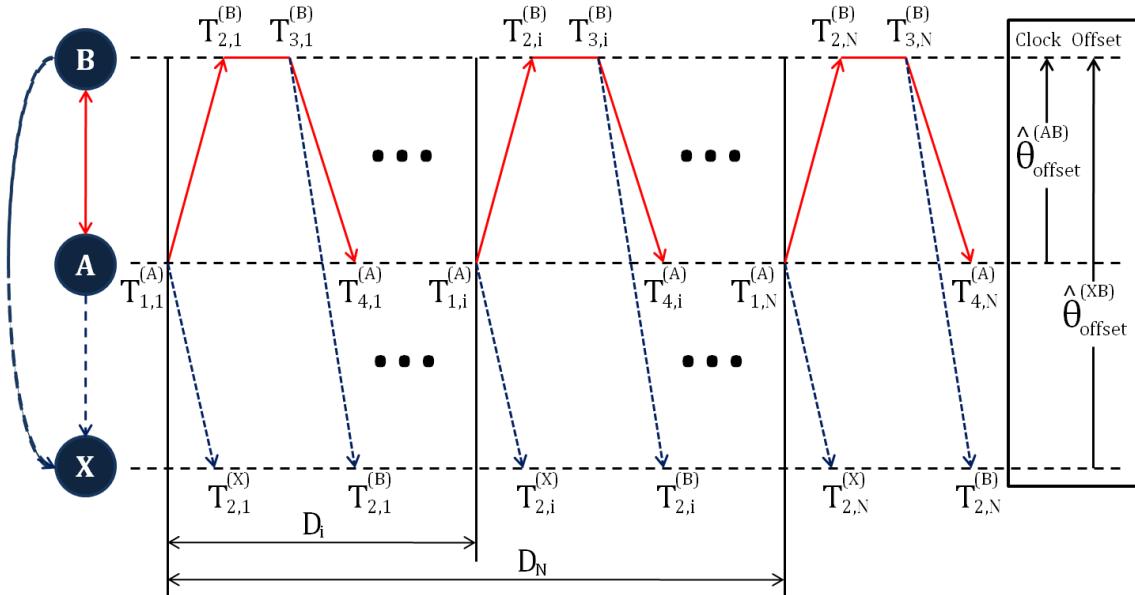


Figure III.4 L'échange de messages pour le protocole PBS.

Dans l'exemple de la Figure III.4, les valeurs d'estampillage $T_{2,i}^{(B)} T_{2,i}^{(B)}$ et $T_{4,i}^{(A)} T_{4,i}^{(A)}$ sont calculées comme indiqué dans les équations suivantes [57]:

$$\begin{aligned} T_{2,i}^{(B)} &= T_{1,i}^{(A)} + \theta_{\text{offset}}^{(AB)} + d^{(AB)} + X_i^{(AB)} \\ T_{4,i}^{(A)} &= T_{3,i}^{(B)} + \theta_{\text{offset}}^{(BA)} + d^{(BA)} + X_i^{(BA)} \end{aligned} \quad (\text{III.2})$$

où $d^{(AB)}$ et $d^{(BA)}$ sont des valeurs fixes (déterministes) pour le retard de propagation dans un lien, et $X_i^{(AB)}$ et $X_i^{(BA)}$ représentent les variations (aléatoires) du retard.

Ainsi, l'estampillage $T_{2,i}^{(B)}$ pour le nœud B, après la prise en compte à la fois du décalage d'horloge (l'offset) et de la valeur de déviation (skew), devient :

$$T_{2,i}^{(B)} = T_{1,i}^{(A)} + \theta_{\text{offset}}^{(AB)} + \theta_{\text{skew}}^{(AB)} * (T_{1,i}^{(A)} - T_{1,1}^{(A)}) + d^{(AB)} + X_i^{(AB)} \quad (\text{III.3})$$

où $\theta_{\text{skew}}^{(AB)}$ représente la déviation relative d'horloge entre le nœud A et le nœud B.

De la même façon, l'horodatage $T_{2,i}^{(X)}$ pour le nœud X, est donnée par l'équation:

$$T_{2,i}^{(X)} = T_{1,i}^{(A)} + \theta_{\text{offset}}^{(AX)} + \theta_{\text{skew}}^{(AX)} * (T_{1,i}^{(A)} - T_{1,1}^{(A)}) + d^{(AX)} + X_i^{(AX)} \quad (\text{III.4})$$

En soustrayant les deux dernières équations (III.3 et III.4), nous obtenons :

$$T_{2,i}^{(B)} - T_{2,i}^{(X)} = \theta_{\text{offset}}^{(XB)} + \theta_{\text{skew}}^{(XB)} * (T_{1,i}^{(A)} - T_{1,1}^{(A)}) + d^{(AB)} - d^{(AX)} + X_i^{(AB)} - X_i^{(AX)} \quad (\text{III.5})$$

Après quelques manipulations mathématiques, le décalage d'horloge et la déviation relative sont donnés par l'équation suivante :

$$\begin{bmatrix} \hat{\theta}_{\text{offset}}^{(XB)} \\ \hat{\theta}_{\text{skew}}^{(XB)} \end{bmatrix} = \frac{1}{N \sum_{i=1}^N D_i^2 - [\sum_{i=1}^N D_i]^2} \times \begin{bmatrix} \sum_{i=1}^N D_i^2 \sum_{i=1}^N x[i] - \sum_{i=1}^N D_i \sum_{i=1}^N [D_i \cdot x[i]] \\ N \sum_{i=1}^N [D_i \cdot x[i]] - \sum_{i=1}^N D_i \sum_{i=1}^N x[i] \end{bmatrix} \quad (\text{III.6})$$

où $D_i \triangleq T_{1,i}^{(A)} - T_{1,1}^{(A)}$, $x[i] = T_{2,i}^{(B)} - T_{2,i}^{(X)}$.

Nous rappelons que la précision de la synchronisation pour le protocole PBS est d'environ $11\mu\text{s}$, comme montrée dans [57]. Du point de vue énergétique, le protocole PBS exige seulement $2N$ messages de synchronisation pour chaque période de synchronisation, où N est le nombre d'échanges de messages, c'est à dire $N_{\text{PBS}} = 2N$. Par conséquent, N_{PBS} ne dépend pas du nombre de capteurs dans le réseau, ce qui réduit considérablement la quantité d'énergie consommée.

Cette dernière observation est d'une importance majeure dans un réseau de capteurs sans fil, ce qui nous a motivé à réfléchir à une solution hybride, capable de répondre aux exigences d'une application comme la nôtre.

III.3 Présentation de la contribution: l'IEEE1588-PBS hybride

En tenant compte de la conception de chacun des deux protocoles présentés ci-dessus et en prenant en considération la spécificité des applications visées, nous arrivons à la conclusion qu'il est nécessaire d'élaborer un protocole de synchronisation fondé sur les avantages précédemment identifiés.

Dans cette partie, nous proposons donc une extension de l'IEEE-1588 qui améliore le rendement énergétique, tout en gardant les performances pour la précision de la synchronisation. Dans l'approche que nous allons proposer, un groupe de nœuds se synchroniseront en écoutant les messages d'une paire de nœuds Maître-Esclave (M-S), selon les principes du protocole PBS précédemment détaillé.

Nous allons démontrer cette approche pour un groupe de 3 nœuds, comme le montre la Figure III.5 :

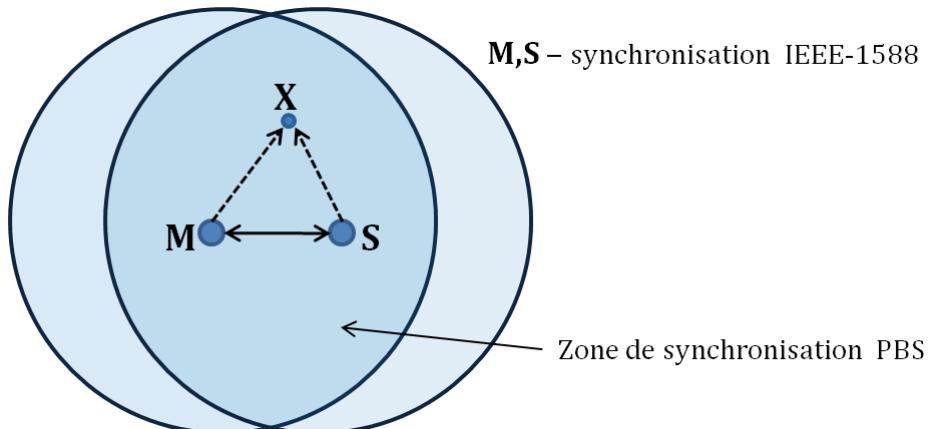


Figure III.5 Schéma de synchronisation.

Le Maître (M) et l'Esclave (S) se synchronisent à travers le protocole IEEE-1588, tandis que le troisième nœud (X) se synchronise en utilisant le principe du protocole PBS. Le nœud X est supposé être à portée directe de M et S car il est juste à l'écoute, pour synchroniser son horloge interne avec celle du nœud M à travers les messages reçus.

III.3.1 Algorithme et fonctionnement

Ici, notre objectif est de présenter le mode de fonctionnement et l'algorithme pour l'extension du protocole de synchronisation IEEE-1588.

La façon dont le nouveau protocole fonctionne est expliquée dans l'exemple suivant (Figure III.6), pour un groupe réduit à seulement 3 nœuds comme annoncé précédemment.

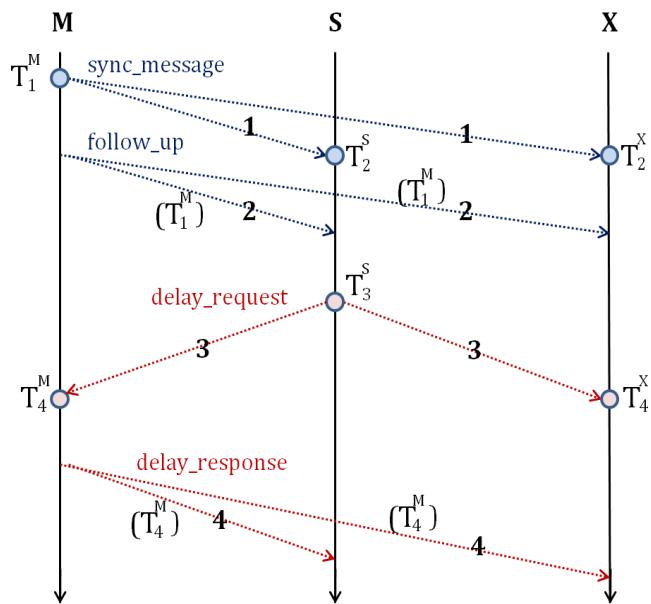


Figure III.6 L'échange des messages dans l'IEEE1588-PBS hybride.

La Figure III.6 montre l'échange des messages entre les éléments du réseau, pour un cycle de synchronisation. L'algorithme est le suivant :

- Le nœud Maître (M) va diffuser un message *sync_message* (1) vers tous les nœuds et va stocker l'heure d'envoi du message (T_1^M). Ensuite, le Maître va émettre le message *follow_up* (2), contenant le précédent estampillage T_1^M .

- b) Le premier paquet de synchronisation arrive à l'Esclave (S) et au niveau du nœud PBS (X), et les deux vont également stocker les temps de réception du message 1, respectivement T_2^S et T_2^X .
- c) Dans la prochaine étape, l'Esclave diffuse au moment T_3^S un message de demande $delay_request$ (3). Ceci est reçu par le nœud M à l'instant T_4^S , ainsi que par le nœud X à $t = T_4^X$. Le Maître va répondre à tout le monde avec le message 4, $delay_response$, qui contient l'information d'horodatage T_4^M .

À la fin de la session de synchronisation, le nœud X détient un certain nombre d'informations ($T_1^M, T_2^X, T_4^X, T_4^S$), contenues dans les messages de diffusion reçus.

Nous notons Δ_{XM} le décalage d'horloge, ainsi que d_{XM} le retard de propagation entre les nœuds X et M et nous supposons que les messages envoyés par l'Esclave sont arrivés au niveau du M et de X en même temps (ce qui implique que $d_{SM} - d_{SX} = 0$). Dans ces conditions, le décalage (Δ_{XM}) et le temps de propagation (d_{XM}) entre les nœuds M et X sont:

$$\begin{aligned}\Delta_{XM} &= T_4^X - T_4^M \\ d_{XM} &= T_2^X - T_1^M - \Delta_{XM}\end{aligned}\tag{III.6}$$

Selon la Figure III.6, les calculs nécessaires pour synchroniser le nœud X du réseau sont donnés par les équations suivantes:

$$\begin{aligned}T_2^X &= T_1^M + \Delta_{XM} + d_{XM} \\ T_4^M &= T_3^S + \Delta_{SM} + d_{SM} \\ T_4^X &= T_3^S + \Delta_{SX} + d_{SX} \\ \Delta_{XM} &= \Delta_{XS} + \Delta_{SM} \\ T_4^M - T_4^X &= \Delta_{XM} + d_{SM} - d_{SX}\end{aligned}\tag{III.7}$$

Face à une vraie application, nous proposons d'intégrer l'algorithme décrit ci-dessus dans le réseau qui a été présenté au long du deuxième chapitre. Pour cela, nous prendrons comme base de travail l'organisation hiérarchique de ce système.

III.3.2 Intégration dans notre cas d'étude

Comme mentionné précédemment, notre réseau de capteurs sans fil est composé d'un grand nombre d'éléments (146 éléments actifs sur les deux ailes), qui doivent communiquer au cours des différentes phases de tests en vol, afin de centraliser les informations de pression et de température dans le cockpit.

Pour cette organisation hiérarchique, nous rappelons les trois catégories d'éléments du réseau : le concentrateur, les routeurs et les nœuds, comme le montre la figure suivante :

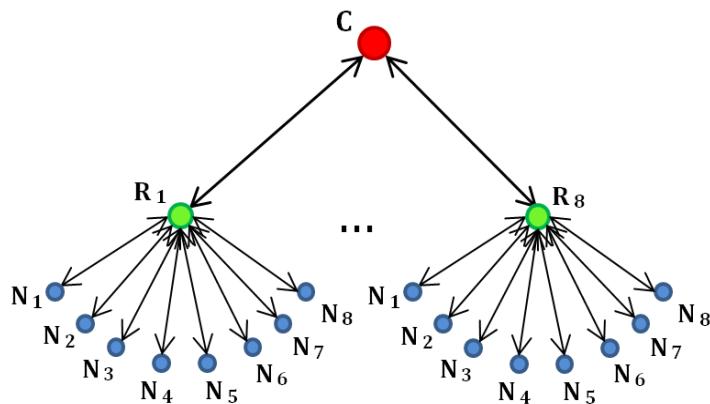


Figure III.7 L'architecture hiérarchique du réseau par aile.

Ainsi, en ce qui concerne le trafic utile, le concentrateur est conçu pour rassembler toutes les informations de la part des routeurs et les transmettre dans la cabine. A leur tour, chacun des huit routeurs recueillent les informations auprès de ses huit nœuds.

Pour le trafic de synchronisation, le concentrateur (C_{1588}) communique avec les routeurs (les éléments R_{1588}), en échangeant des messages conformément au principe de l'IEEE 1588 (Niveau 1 de synchronisation) (Figure III.8).

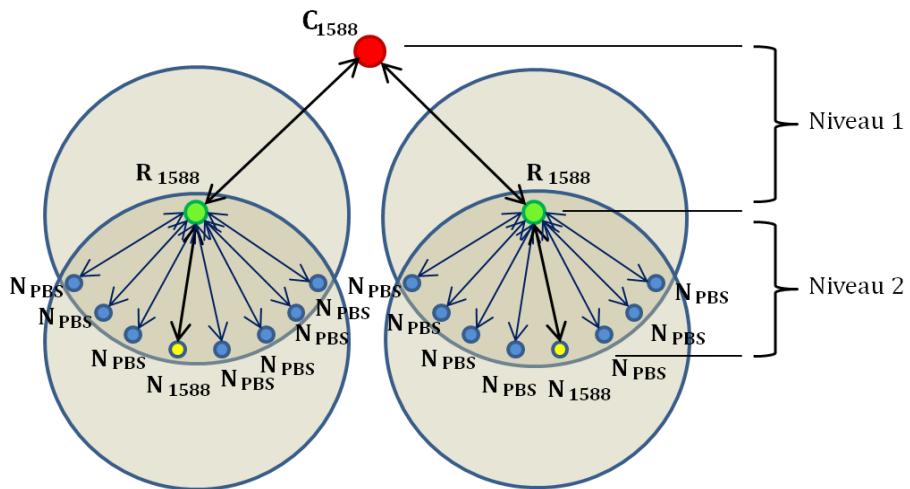


Figure III.8 Les niveaux de synchronisation dans le réseau.

Dans un groupe composé d'un routeur et de ses huit nœuds (Niveau 2 de synchronisation), nous avons :

- ❖ Un nœud central (N_{1588}) qui se synchronisera avec le routeur (R_{1588}) à travers le même principe utilisé pour le Niveau 1.
- ❖ Les autres sept nœuds (N_{PBS}) qui se synchroniseront en utilisant les informations contenues dans les messages reçus, en concordance avec l'algorithme de l'IEEE1588-PBS hybride présenté auparavant (Figure III.6), et illustré par la figure suivante (Figure III.9) :

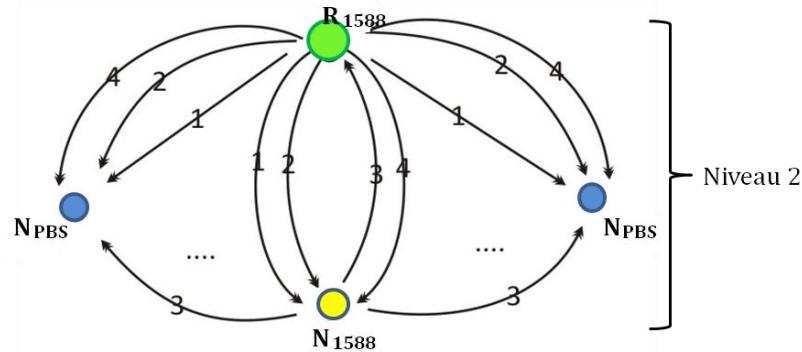


Figure III.9 Les messages échangés dans le Niveau 2 de synchronisation.

La condition qui s'impose pour chaque groupe de Niveau 2, est que tous les nœuds doivent être à la portée directe de leur routeur (R_{1588}) et du nœud central (N_{1588}), de sorte que les nœuds N_{PBS} recevront tous les messages échangés entre la paire R_{1588} - N_{1588} .

Donc, nous pouvons constater que cet algorithme de synchronisation reste cohérent avec notre application et avec la topologie réseau que nous avons présentée auparavant.

III.3.3 L'impact sur la consommation d'énergie

En ce qui concerne le bilan énergétique, compte tenu du nombre de messages de synchronisation diffusés entre le routeur et le nœud central, théoriquement ils vont consommer plus d'énergie que les nœuds N_{PBS} , qui sont seulement à l'écoute. Cet aspect (l'écoute) est un grand avantage pour un réseau composé de centaines d'éléments, car cela entraîne une importante économie d'énergie globale, tout en gardant les performances de synchronisation.

Concernant cette dernière observation, nous reviendrons dans le chapitre « Simulations et expérimentation » avec des explications et des mesures.

III.4 **Implémentation et mise en œuvre**

Dans les sections qui suivent nous présenterons les conditions/étapes de la mise en œuvre matérielle et de l'implémentation logicielle de notre protocole de synchronisation.

III.4.1 Conditions pratiques de mise en œuvre

La mise en œuvre pratique du protocole IEEE1588-PBS hybride demande, pour pouvoir garantir les performances annoncées préalablement, un certain nombre d'exigences matérielles.

Ainsi, les nœuds physiques doivent être équipés, par exemple, d'un eTPU (Enhanced Time Processing Unit) avec 32 canaux d'entrée et une horloge sur 24 bits, qui est une unité d'horodatage très précise. La spécificité réside dans l'horloge du processeur qu'est contrôlée par un quartz, qui, idéalement oscille à une fréquence constante. Par contre, dans le monde réel, l'oscillateur a une tolérance qui se traduit par une dérive de la fréquence (un oscillateur avec une tolérance en fréquence d'une partie par million (ppm) a une vitesse de dérive de l'ordre de la microseconde par seconde).

En général, un oscillateur à bas prix a une tolérance en fréquence allant de 20 à 50 ppm, avec une plage pour le taux maximum de la dérive de 20 à 50 μ s par seconde. Pour atteindre une précision de synchronisation souhaitée dans notre application, il est nécessaire que l'horloge du processeur soit commandée par un TCXO³⁶ (Temperature Compensated Crystal Oscillator). L'avantage est qu'il utilise une tension de correction, qui va réduire la variation de la fréquence en température. Plus exactement, la tension de correction va être appliquée à une diode du circuit, de telle sorte que la fréquence du cristal va avoir une variation très petite (une tolérance qui peut s'approcher de 0.1 ppm).

Enfin, tester réellement un tel protocole nécessite tout d'abord une étape de validation par simulation. Pour cela, nous allons présenter dans la section suivante son algorithme d'implémentation dans le logiciel NS-2.

III.4.2 Les étapes d'implémentation dans le simulateur NS-2

NS-2 est un outil de simulation très complexe, qui a l'avantage d'être un logiciel open source. Cet aspect nous permet d'intégrer et de développer notre algorithme de synchronisation à l'aide de nouveaux objets C++, et d'utiliser l'interface de configuration OTcl pour faire fonctionner ensemble ces objets.

Par la suite, nous présenterons donc l'implémentation d'un modèle d'horloge interne, étape obligatoire qui permettra de simuler la dérive individuelle de chaque nœud du réseau. Ensuite, nous continuerons avec l'intégration du protocole IEEE-1588 dans le NS-2, en l'adaptant aux conditions imposées par notre application, tout en restant générique dans la manière d'implémenter. En utilisant la même stratégie, nous mettrons en œuvre le nouveau protocole de synchronisation proposé: IEEE1588-PBS hybride.

III.4.2.1 L'intégration du modèle d'horloge interne

Le simulateur de réseau NS-2 dispose d'une horloge de simulation globale, en fonction de laquelle tous les événements sont programmés et exécutés. Pour la mise en œuvre d'un protocole de synchronisation, nous devrons tous d'abord implémenter un modèle d'horloge individuelle pour chaque nœud. Ainsi, chaque horloge doit fonctionner différemment par rapport aux autres, afin de pouvoir simuler des réseaux réels.

Sous NS-2, nous avons identifié plusieurs méthodes pour implémenter une horloge interne qui soit individuelle pour chaque nœud du réseau. Mais, pour pouvoir obtenir une meilleure précision, nous avons intégré le module d'horodatage des messages au niveau de la couche physique.

Plus précisément, nous avons apporté des modifications au niveau des fichiers source «*wireless-phy.cc&.h*» et nous avons crée deux nouvelles commandes de langage Tcl, «*attach_clock*» et «*alloc_clock*», qui seront appelées pour chaque nœud de la simulation. En appelant ces commandes, chaque élément aura une horloge indépendante qui dérivera d'une manière similaire à un oscillateur réel.

Comme mentionnées dans la partie III.2.4, nous prenons comme référence les conditions annoncées par les auteurs du [53] dans leurs expérimentations. Ainsi, nous avons simulé un oscillateur en cristal avec compensation en température (TCXO) qui a une

³⁶ <http://www.vectron.com/products/tcxo/tx-600.pdf>

tolérance en fréquence de 1,5 ppm, qui correspond à une dérive d'horloge de 1,5 μ s pour chaque seconde.

Pour obtenir cette valeur de dérive des horloges, nous avons utilisé la fonction «Random» de NS-2 qui nous a permis d'incrémenter nos horloges de sorte que, après une seconde, la dérive soit d'environ 1,5 microseconde.

Pour une meilleure analyse du fonctionnement de notre protocole de synchronisation et une meilleure simulation d'un réseau désynchronisé, nous avons décidé de démarrer chaque horloge individuelle avec une dérive aléatoire de l'ordre de quelques millisecondes. Cette mise en œuvre nous permettra de savoir plus exactement, combien de temps, il faudra à l'ensemble du réseau pour se synchroniser.

III.4.2.2 La phase d'implémentation d'IEEE-1588

Dans cette section, nous allons présenter les étapes d'implémentation pour le protocole de synchronisation IEEE-1588 sous NS-2 et les modifications apportées aux fichiers sources pour simuler le fonctionnement de ce protocole dans notre architecture de communication.

Nous avons tout d'abord commencé par la définition d'un nouveau type de paquet, spécifique pour le trafic de données de notre application. L'en-tête de ce nouveau paquet contient hormis les champs spécifiques pour notre trafic de données, un champ dédié à la synchronisation (l'estampillage du paquet reçu et transmis) et un identificateur spécial qui sert à distinguer les différents types de paquets utilisés par le protocole de synchronisation.

Nous rappelons que notre réseau se compose d'un concentrateur, huit routeurs et huit nœuds associés à chaque routeur (par aile). Comme nous l'avons déjà expliqué dans la partie implémentation de la topologie réseau, pour assurer une communication hiérarchique, nous avons imposé un certain nombre de restrictions de communication qui sont réalisées par un filtrage du trafic au niveau de la couche MAC. A ce sujet, nous avons créé une nouvelle commande Tcl nommée cette fois-ci «alloc_ip_sync», qui nous permettra d'analyser le trafic de synchronisation indépendamment du trafic utile.

Nous allons décrire par la suite les étapes pour l'intégration de notre protocole de synchronisation dans l'outil de simulation NS-2, ainsi que les autres modifications qui ont été nécessaires.

L'IDENTITE CONCENTRATEUR

Pour implémenter le protocole souhaité avec NS-2 et en même temps pour le rendre compatible avec notre architecture réseau, nous avons créé pour le concentrateur un nouvel agent «sync_agent_master_con» et une nouvelle application «sync_app_master», en respectant les principes d'implémentation décrits dans le chapitre précédent.

En ce qui concerne les niveaux de synchronisation (Figure III.8), le concentrateur doit se synchroniser avec les huit routeurs pour lesquels il a le rôle de «maître», présenté dans la description de l'algorithme (Figure III.1, Figure III.2). La phase de synchronisation est démarrée par la nouvelle application à un instant t , spécifiée dans le script Tcl. Cette phase commence par l'envoi du premier message de synchronisation (*sync*), en déclarant l'identifiant du paquet «*pkt_no*» à «0». A l'instant immédiat, le message *follow_up* est créé et identifié à «1».

Les deux messages vont passer du niveau applicatif au niveau agent (couche transport) en appelant des fonctions spécifiques. L'agent va ainsi horodater les deux paquets au niveau de la couche physique et ensuite les diffuser à tous les routeurs du réseau.

Lors de la réception du message *delay_request* (Figure III.2) qui a l'identifiant «*pkt_no = 2*», chaque routeur va mémoriser au niveau de la couche physique le moment de réception du paquet. Par la suite, l'agent transmet les informations contenues dans l'en-tête de paquet à la couche application, où une réponse est créée avec l'identifiant «*pkt_no = 3*» et transmis de nouveau à l'agent, qui transmettra les informations individuellement à chaque routeur.

Après avoir présentés les fonctionnements accomplis par le concentrateur dans le processus de synchronisation, dans le paragraphe suivant nous décrivons pas à pas les rôles joués par le routeur dans cet algorithme de synchronisation.

L'IDENTITE ROUTEUR

Pour la suite de la mise en œuvre du protocole IEEE-1588, nous avons créé pour les routeurs deux nouveaux agents: le «*sync_agent_slave_router*» qui aura le rôle «d'esclave» dans le Niveau 1 de synchronisation et le «*sync_agent_master*» qui servira de «maître» pour le Niveau 2 de synchronisation (Figure III.8).

Nous avons également créé deux nouvelles applications : la «*sync_app*» qui sera reliée à l'agent «*sync_agent_slave_router*», tous les deux responsables du trafic de synchronisation entre le concentrateur et les routeurs, ainsi que du trafic de données utiles. La deuxième application, l'«*app_master_router*» sera reliée à l'agent «*sync_agent_master*», qui seront impliqués seulement dans le trafic de synchronisation entre les routeurs et leurs nœuds associés.

En ce qui concerne le trafic de synchronisation, le routeur va atteindre deux principaux rôles: «d'esclave» pour le Niveau 1 de la communication et de «maître» pour le Niveau 2.

Dans le Niveau 1 de synchronisation, les routeurs reçoivent de la part du concentrateur les messages *sync* et *follow_up*, dont le premier est horodaté à son arrivée au niveau physique. Cette information de temps est ainsi envoyée à la couche application, en respectant l'algorithme d'IEEE-1588.

Lorsque les informations contenues dans le message *follow_up* atteignent le niveau applicatif, la précision de la synchronisation est instantanément vérifiée. Si la valeur est inférieure à 500 nanosecondes, cela signifie que les nœuds concernés sont synchronisés et ce cycle de synchronisation s'arrête ici.

Sinon, le routeur envoie directement au concentrateur un message *delay_request* horodaté à son départ et attend le message *delay_response* de la part du concentrateur. Une fois la réponse reçue, la couche application calcule le décalage et le délai de propagation. Ainsi, au démarrage de la prochaine session de synchronisation, les horloges internes des routeurs seront modifiées selon les valeurs calculées.

Dans le Niveau 2 de synchronisation, chaque routeur agit comme un « maître » pour l'ensemble des huit nœuds qui lui sont associés. Le cycle commence par la diffusion des messages *sync* et *follow_up*, conformément au protocole IEEE-1588. Ensuite, le routeur

attend le message `delay_request` et répond à chaque nœud avec un message `delay_response` (similaire au fonctionnement du concentrateur).

Pour assurer une synchronisation pour l'ensemble de notre système, il nous reste à définir les fonctions accomplies par les nœuds.

L'IDENTITE NŒUD

Pour chaque groupe de huit nœuds associés aux routeurs, nous avons implémenté un nouvel agent «`sync_agent_slave`» (couche transport). Pour le niveau applicatif, nous avons pu utiliser la même application mise en place pour les routeurs, «`sync_app`», car les nœuds suivent le même algorithme.

Ainsi, les nœuds s'identifient comme des «esclaves» et suivent exactement les mêmes étapes que celles présentées précédemment pour les routeurs dans le Niveau 2 de synchronisation.

Dans la figure suivante (Figure III.10), nous avons représenté les principales modifications que nous avons apportées au niveau de la pile protocolaire du logiciel NS-2, dans le but de simuler le protocole de synchronisation IEEE-1588 :

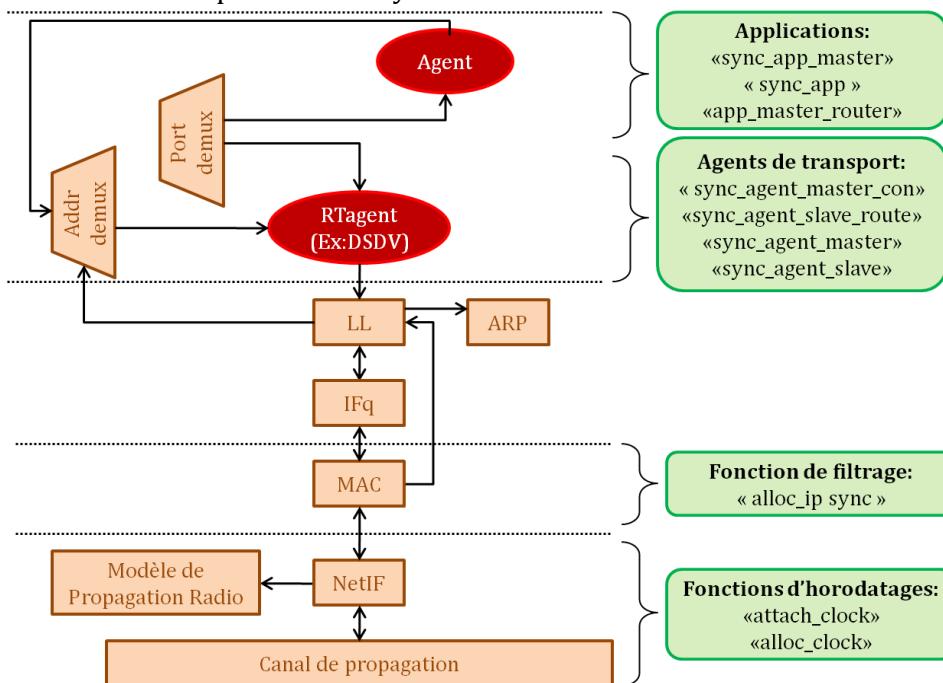


Figure III.10 Intégration de l'IEEE-1588 dans l'architecture du logiciel NS-2.

Une fois décrit le fonctionnement de chaque identité (concentrateur, routeur, nœud) de notre réseau, nous devrons rajouter que le système sera intégralement synchronisé quand les deux niveaux de communication (Niveau 1 et Niveau 2) seront déjà synchronisés indépendamment l'un de l'autre.

III.4.2.3 La phase d'implémentation d'IEEE1588-PBS hybride

Dans cette section, nous présentons l'implémentation de notre proposition de synchronisation: l'IEEE1588-PBS hybride.

Comme nous l'avons mentionné précédemment dans la présentation du fonctionnement de notre protocole, dans un groupe de N nœuds, il y aura toujours deux supers nœuds qui se synchroniseront en suivant les règles de la norme IEEE-1588. Le reste

des nœuds se synchroniseront à travers les messages diffusés par les deux nœuds spéciaux, en utilisant les informations existantes dans les messages reçus (notamment les estampillages effectués au niveau de la couche physique). Dans la topologie de notre réseau, nous appliquerons cette théorie pour le Niveau 2 de communication (Figure III.8).

En ce qui concerne le Niveau 1, les éléments (concentrateur et les routeurs associés) se synchroniseront en utilisant le principe d'IEEE-1588. Le choix de cette stratégie est motivé pour les raisons suivantes:

- ❖ Compte tenu du fait que le concentrateur et les routeurs seront potentiellement alimentés, l'utilisation d'un protocole gourmand en énergie ne posera pas de problème sur la durée de vie globale de notre système ;
- ❖ De l'autre côté, l'utilisation d'un protocole d'une très grande précision (dizaines, centaines de nanosecondes) permettra dans notre contexte de tirer avantage d'une très bonne synchronisation pour le Niveau 1 de communication, qui va indirectement influencer les performances de synchronisation pour le Niveau 2 (les routeurs seront déjà synchrones).

Par conséquent, le fonctionnement du concentrateur et des routeurs sera le même que celui décrit dans la section précédente, et seule l'implémentation des nœuds sera différente.

Ainsi, dans le Niveau 2 de communication, le routeur se synchronisera avec le nœud central nommé N_{1588} (Figure III.8) en utilisant le principe de l'IEEE-1588. Les autres sept nœuds nommés N_{PBS} , recevront les messages échangés entre la paire routeur- N_{1588} . Nous devons mentionner que le nœud N_{1588} diffuse le message `delay_request` et ne l'envoie pas directement au routeur comme dans la mise en œuvre du protocole IEEE-1588.

L'IDENTITE NŒUD N_{PBS}

Pour les nœuds de type N_{PBS} nous avons conçus un nouvel agent «`sync_agent_pbs`» et une nouvelle application «`sync_app_pbs`». Les messages échangés entre la paire routeur- N_{1588} sont également reçus par les agents attachés aux nœuds N_{PBS} et horodatés au niveau de la couche physique. Ces paquets arriveront en suite à la couche applicative, où le traitement des informations suit l'algorithme présenté auparavant (Figure III.10).

Nous rappelons qu'un nœud N_{PBS} va seulement recevoir les messages de synchronisation diffusés par les deux «super nœuds», sans avoir une participation active dans ce trafic. Cet aspect va nous permettre de réduire considérablement la consommation d'énergie au niveau des nœuds N_{PBS} : cette hypothèse sera démontrée dans le dernier chapitre de ce manuscrit.

Dans la Figure III.11, nous présentons les modifications qui nous ont permis d'intégrer le protocole de synchronisation IEEE1588-PBS hybride dans le logiciel de simulation NS-2.

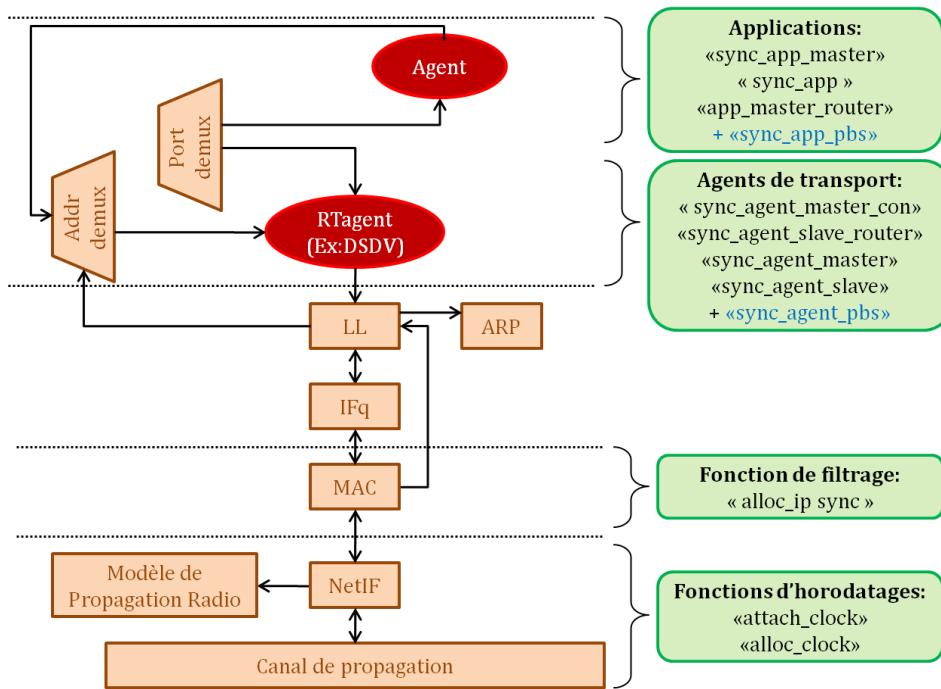


Figure III.11 Intégration du l'IEEE1588-PBS hybride dans l'architecture du logiciel NS-2.

L'implémentation de ces deux protocoles de synchronisation dans le même simulateur va nous permettre de réaliser une analyse cohérente de leurs performances, qui va servir comme base de comparaison pour montrer les avantages apportés par notre solution.

III.5 Conclusion

Ce chapitre s'est focalisé sur la proposition d'un nouveau protocole de synchronisation qui se doit d'être performant du point de vue précision de la synchronisation, ainsi que du point de vue énergétique.

Notre premier objectif a été d'étudier les protocoles de synchronisation existants dans la littérature. Après avoir analysé plusieurs solutions, nous avons conclu qu'aucune n'est capable de répondre à l'ensemble des exigences imposées par notre application. Ainsi, nous avons pris en compte deux algorithmes : celui de la norme IEEE-1588 qui a l'avantage d'être le meilleur du point de vue précision de la synchronisation et celui du protocole PBS qui a l'avantage d'être très économique en énergie.

L'objectif suivant a été donc de développer un algorithme de fonctionnement basé sur ces deux protocoles, qui permettra de réduire la consommation d'énergie sans altérer les performances en terme de synchronisation de l'IEEE-1588.

Nous avons donc décidé d'implémenter notre proposition, ainsi que la norme IEEE-1588 dans l'outil de simulation NS-2, pour pouvoir faire une comparaison entre les deux solutions (l'IEEE-1588 et l'IEEE1588-PBS hybride) et démontrer les avantages de la nouvelle proposition en termes de précision pour la synchronisation et de consommation en énergie. Pour cela, nous avons mis en œuvre aussi une horloge interne avec une dérive de 1,5 microseconde par seconde pour chaque noeud, afin de simuler des conditions réelles.

Les résultats ainsi obtenus vont être présentés et analysés en détails dans le Chapitre V (Simulations et expérimentations).

IV LES ANTENNES DIRECTIVES DANS LES RESEAUX DES CAPTEURS SANS FIL

| | | |
|-------------|--|------------|
| IV.1 | Introduction | 101 |
| IV.2 | L'impact des antennes directives sur les différentes couches de communication | 103 |
| IV.2.1 | Introduction | 103 |
| IV.2.2 | Concepts de base | 104 |
| IV.2.3 | Propositions pour l'utilisation d'antennes directives | 107 |
| IV.3 | L'applicabilité dans une catégorie large de réseaux sans fil..... | 113 |
| IV.4 | Principe et implémentation de notre solution..... | 114 |
| IV.4.1 | Proposition d'architecture d'antennes directives | 115 |
| IV.4.2 | L'implémentation de notre solution | 117 |
| IV.5 | Conclusion | 118 |

Ce chapitre a pour but de traiter l'aspect des antennes directives ou intelligentes dans les réseaux d'instrumentation. Dans un premier temps, nous présenterons une étude sur les technologies et les solutions déjà existantes, en cherchant à identifier les avantages apportés par ces proposition, notamment sur l'aspect consommation d'énergie et la qualité de service. Ces dernières nous serviront de base pour la conception d'une solution qui correspond à nos besoins. Enfin, nous terminerons ce chapitre par l'implémentation de notre modèle d'antennes directives dans l'outil de simulation NS-2, ce qui nous permettra de l'intégrer dans l'architecture de communication spécifique à notre application.

IV.1 Introduction

La terminologie de transmission sans fil est directement liée à la notion d'antenne. Cet élément est basé sur un conducteur électrique d'une complexité plus ou moins élevée, permettant de rayonner ou de capter des ondes électromagnétiques. Autrement dit, l'antenne permet de transformer un signal électrique en une onde électromagnétique pour l'émission et inversement pour la réception [112].

Les paramètres qui définissent une antenne sont nombreux (Tableau IV.1).

La directivité est une caractéristique importante dans le choix d'une antenne. C'est dans ce contexte que nous allons tracer le diagramme de rayonnement, qui nous permet de visualiser les lobes de rayonnement dans les trois dimensions.

| Paramètres | Précisions |
|----------------------------------|---|
| Fréquence(s) de fonctionnement | MHz, GHz, ... |
| Bandé passante | MHz, ... |
| Polarisation | Verticale, horizontale, ... |
| Impédance | 50, 75, 300 Ohms, ... |
| Directivité | Le diagramme de rayonnement |
| Forme et dimensions | Très grandes variétés |
| Mode d'alimentation | Ligne bifilaire, câble coaxial, guide d'onde, ... |
| Gain | dBi ou dBd |
| Caractéristiques mécaniques | Dimensions, masse, ... |
| Puissance maximale d'utilisation | W, dBm, ... |

Tableau IV.1 Les paramètres généraux des antennes.

Ainsi, la Figure IV.1 présente les deux grandes catégories de diagrammes de rayonnement (représentés en plan horizontal): pour des antennes omni/bidirectionnelles (a) et pour des antennes directives (b). La différence entre ces deux représentations est que dans le diagramme de gauche (a) les antennes rayonnent dans toutes les directions, contrairement à l'antenne du diagramme de droite (b) qui ne va émettre que dans une seule direction, donnée par l'orientation du lobe principal.

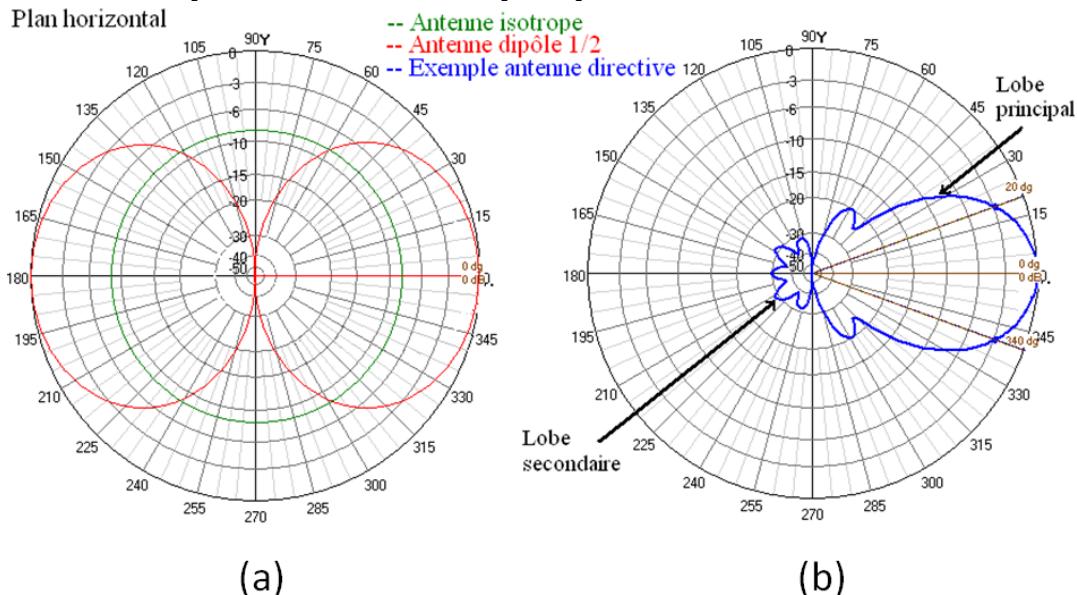


Figure IV.1 Les diagrammes de rayonnement pour différentes antennes.

Le *gain* d'une antenne représente la quantité d'énergie (émise ou reçue) dans une direction. Pour mieux comprendre la notion de directivité des antennes, l'analogie peut être faite avec l'énergie lumineuse d'une bougie qui peut être concentrée grâce à une lentille convergente. Il y a deux possibilités de calcul du gain : l'exprimer en dBi par rapport à l'antenne isotrope³⁷ (idéale) ou en dBd par rapport à l'antenne dipôle 1/2 onde³⁸.

³⁷ L'antenne isotrope a toujours le gain de 0dBi.

³⁸ L'antenne dipôle 1/2 onde possède le gain de 2, 15 dBi soit 0dBd.

La classification des antennes la plus générale est en lien étroit avec les paramètres précédemment évoqués qui nous permettent de parler de deux grandes classes d'antennes :

- ❖ Les antennes omnidirectionnelles : elles rayonnent de la même façon dans toutes les directions (antenne isotrope ou idéale). En réalité, n'importe quelle antenne subit l'influence de l'espace dans lequel elle est placée : sol, propre support, etc.;
- ❖ Les antennes directives : elles permettent d'atténuer les émissions/parasites qui viennent d'autres directions que celle souhaitée. Dans ce cas, nous allons avoir un ou deux lobes principaux et/ou des lobes secondaires (Figure IV.1(b)).

L'exemple présenté dans la Figure IV.1(b) montre qu'une antenne dite « directive », en réalité n'est pas 100% directive ; les preuves sont les lobes parasites (secondaires) situés dans tout le secteur arrière y compris dans l'axe du lobe principal. Cet effet est pris en compte par des paramètres comme « le rapport Avant/Arrière », « le rapport Avant/Côtés » ainsi que par le diagramme de rayonnement de l'antenne dans le plan horizontal [112].

La propriété d'émission dans toutes les directions pour une antenne omnidirective implique des inconvénients, comme un gain plus faible par rapport à celui d'une antenne directive, ou l'effet de capter des signaux indésirables. Nous pouvons trouver aussi des antennes bi-directives, autrement dit des antennes qui rayonnent autant vers l'avant que vers l'arrière, comme l'antenne dipôle 1/2 (Figure IV.1 (a)).

Le principe de fonctionnement d'une antenne directionnelle consiste donc à utiliser un faisceau focalisé sur une direction précise avec une largeur de faisceau prédéfinie. Cette technologie présente plusieurs avantages, comme une transmission avec un minimum des sauts, l'augmentation de la communication simultanée ou la sélectivité.

Dans [118], nous trouvons un modèle d'antenne directionnelle conçu pour les réseaux maillés (mesh) sans fil, qui présente des avantages significatifs en termes de débit et retards par rapport au cas de l'antenne omnidirectionnelle.

Les avantages de cette technologie d'antenne sont également exploités dans des applications sans fil sur le corps humain. Ainsi, dans [145], les auteurs présentent un modèle d'antenne basé sur une structure Ultra Wide Band de type fente, qui présente une meilleure efficacité en termes de rayonnement dans le voisinage d'un corps humain.

En fonction de l'application, l'utilisation des antennes directives a des avantages, comme éviter de rayonner dans une zone où il pourrait y exister des interférences avec d'autres émetteurs ou de réaliser une économie d'énergie par rapport à des antennes omnidirectionnelles, mais aussi des inconvénients, en particulier la complexité de gestion induite.

IV.2 L'impact des antennes directives sur les différentes couches de communication

IV.2.1 Introduction

Les recherches dans le domaine des réseaux de capteurs sans fil (WSNs) offrent de plus en plus de solutions orientées vers la fiabilité, l'auto-organisation, la réduction de la consommation d'énergie etc. Ces aspects sont très importants pour un système dans lequel les éléments de communication sont nombreux et limités en énergie (alimentés par batteries). Cela impose, en premier temps, le choix de technologies et de protocoles de

communication qui tiennent compte de toutes les contraintes imposées par chaque application.

L'utilisation d'un nouveau type d'antenne implique un travail d'adaptation de la pile protocolaire du modèle de communication OSI. Plus précisément, les niveaux de communication qui vont souffrir des changements sont :

- ❖ La couche physique qui contient le transducteur RF (Radio Fréquence) et qui fournit une interface entre le niveau MAC et le canal radio ;
- ❖ La couche MAC qui est responsable de l'accès au canal physique ;
- ❖ La couche réseau qui assure les fonctionnalités de routage et de configuration du réseau.

La question est pourquoi devrons-nous modifier aussi le niveau réseau, vu que les antennes sont censées collaborer plutôt avec les couches bases (PHY et MAC). Prenons l'exemple d'une couche physique qui intègre un mécanisme d'antennes intelligentes. A travers une collaboration entre les différents protocoles (de routage, d'accès au medium, de niveau physique, etc.), nous pourrons imaginer de nouveaux mécanismes comme la localisation par triangulation, la transmission synchrone des paquets, la reconfigurabilité dynamique, l'efficacité en terme de débit, etc. Ces structures sont connues sous le nom d'architectures cross-layer et leur principal avantage est de permettre un contrôle optimal du medium de communication.

Par exemple, Arun Ranjikar et al. proposent dans [113] un protocole de routage collaboratif pour des réseaux sans fil multi-sauts qui utilisent l'avantage des antennes directives et de la diffusion sans fil. Le processus de collaboration est soutenu par des informations obtenues de la couche MAC dans le but de former des liens temporaires qui serviront à maintenir la transmission jusqu'à ce que la route détériorée soit réparée ou qu'un nouvel itinéraire soit généré. Une autre proposition intéressante est celle présentée dans [114], où les auteurs introduisent un mécanisme « cross-layer packet scheduler » adaptif qui augmente les performances du réseau, tout en respectant le délai du paquet.

Ce type de stratégie permettra donc de rendre plus efficace le service réseau (réduire les interférences, faire des économies d'énergie, etc.) et d'envisager de nouveaux services, comme la localisation.

IV.2.2 Concepts de base

Parmi les problèmes qui se posent principalement dans un réseau de capteurs, les plus importants sont : la création et le maintien d'une topologie de réseau qui permette un maximum de connectivité, et la réduction de la consommation d'énergie.

Ces deux objectifs peuvent être opposés. Prenons l'exemple d'un ensemble de capteurs qui transmettent leurs informations simultanément. Le risque d'interférences ou de collisions peut causer une perte importante d'informations transmises, de la connectivité, et donc de l'énergie si la retransmission doit avoir lieu. Ainsi, les couches supérieures de transmission doivent développer de nouveaux services qui rendent la gestion des réseaux de capteurs plus efficace.

Nous détaillons dans la suite les principaux problèmes soulevés par l'utilisation d'antennes directives au niveau de la couche liaison des données.

PROBLEME DE NŒUDS CACHES

Un phénomène qui apparaît au niveau MAC, spécifique aux réseaux sans fil, est bien connu sous le nom *de nœud caché*. Il est généralement lié à la distance ou à la présence d'un obstacle. Cette situation s'installe quand deux stations situées de chaque côté d'un point d'accès peuvent entendre toutes les deux une activité du point d'accès, mais pas de l'autre station (Figure IV.2). Ainsi, il est possible que les deux stations transmettent en même temps un message au point d'accès. Pour résoudre ce problème, la norme 802.11 a défini un protocole MAC optimal basé sur l'échange de paquets RTS / CTS [115].

L'inconvénient des antennes directives est qu'en leur présence, ce mécanisme ne devient plus suffisant. Ainsi, la Figure IV.2 décrit une telle situation : les nœuds A et B échangent des paquets RTS/CTS en mode directif ; C ne peut pas écouter la communication entre A et B. Supposons maintenant que C veuille émettre un paquet vers le nœud D. Le nœud C initialise un paquet « Carrier Sensing » dans la direction de D et conclut que le canal est libre, alors C envoie un RTS vers D. Ce paquet va interférer avec les paquets de A à destination de B.

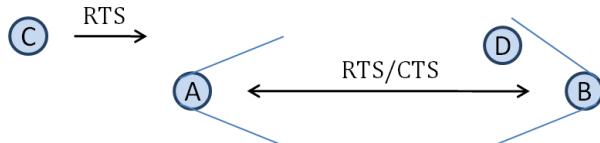


Figure IV.2 Le problème de nœuds cachés.

PROBLEME DE FAMINE

Nous allons présenter dans la suite un autre problème lié à l'utilisation d'antennes directives. C'est le phénomène de famine qui se produit lorsqu'un nœud C essaie d'envoyer un paquet RTS vers un nœud A, alors qu'il est en communication directionnelle avec un autre nœud B (Figure IV.3).

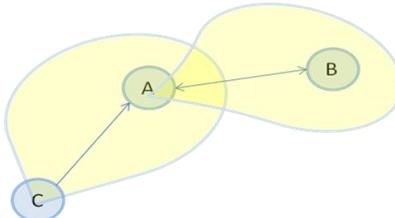


Figure IV.3 Problème de famine dans les réseaux sans fil.

Suite à la non réception du paquet CTS, le nœud C va retransmettre le paquet RTS plusieurs fois et augmenter le « backoff » à chaque retransmission. Après n retransmissions, le nœud C va supprimer le paquet.

Cette situation peut avoir des conséquences lourdes en ce qui concerne la consommation d'énergie, qui est cruciale dans un réseau de capteurs sans fil.

PROBLEME DE CAPTURE

Le phénomène de capture apparaît quand, par exemple, le nœud 0 est capable d'atteindre et de communiquer en mode directionnel avec le nœud 2, et le nœud 4 avec le nœud 1 (Figure IV.4).

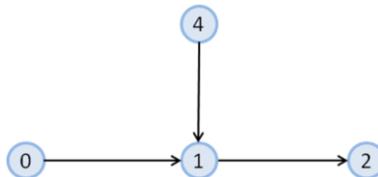


Figure IV.4 Problème de capture dans les réseaux sans fil.

Considérons le moment où tous les nœuds sont en veille et que les nœuds 0 et 4 commencent une communication directionnelle vers leurs destinations (respectivement les nœuds 2 et 1). Si le nœud 0 fait la transmission en premier, à la fois le nœud 2 et le nœud 1 choisiront l'antenne qui pointe vers le nœud 0, et recevront le paquet. Ainsi, le nœud 1 ne serait pas en mesure d'écouter le nœud 4 quand il va initier la communication vers le nœud 1, puisque l'antenne utilisée est déjà choisie (celle orientée vers le nœud 0). Une telle situation va conduire aussi à des pertes inutiles en énergie et va diminuer les performances du système en terme de délais de transmission, ou de connectivité.

LES POINTS FORTS

Les avantages liés à l'utilisation d'antennes directionnelles sont nombreux. Nous avons décidé de mentionner les plus importants ci-dessous.

Dans un contexte de transmission multi sauts, en exploitant le gain de l'antenne directionnelle qui est plus important que pour une antenne omnidirectionnelle, nous pouvons réduire la distance de communication à un nombre minimal de sauts et parfois uniquement à une transmission à long saut. Cet aspect peut contribuer à améliorer le délai de transmission, diminuer le nombre de messages de contrôle, et réduire la congestion du réseau.

L'utilisation d'antennes directives permet à beaucoup plus d'utilisateurs de communiquer en même temps, ce qui contribue à l'augmentation de la capacité du réseau.

Dûe à la capacité d'une antenne directive d'émettre sous un angle spécifique, un nœud peut être en mesure de recevoir sélectivement des signaux d'une direction souhaitée. Cela permet au nœud récepteur d'éviter les interférences, ce qui augmente le rapport signal sur bruit (SNR). Ce paramètre va contribuer à son tour à augmenter le débit et réduire le délai de bout en bout, car les retransmissions sont ainsi évitées. L'évitement des retransmissions entraîne une économie en énergie du système.

Ainsi, basé sur un contrôle des diagrammes d'antennes reconfigurables, les protocoles de routage pourront être optimisés pour limiter les interférences, car une diffusion omnidirectionnelle peut générer des conflits d'accès. C'est l'exemple de la double paire illustrée dans la Figure IV.5 (a), où le nœud B parasites lors de sa communication avec le nœud A, la communication entre les nœuds C et D.

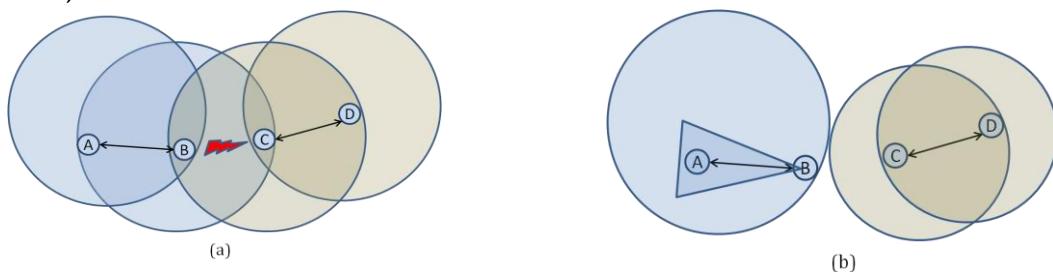


Figure IV.5 Problème de la double paire.

Ce phénomène pourrait être évité, par exemple, par la mise en place d'un diagramme d'antenne directionnelle pour B (Figure IV.5 (b)), ou une réduction de la puissance d'émission qui diviserait le réseau en deux composantes connexes.

Minimiser la consommation d'énergie est un objectif majeur pour les WSNs. Ainsi, dans [139], Di Tian et N. D. Georganas proposent une solution pour l'extension de la durée de vie du réseau. Le procédé consiste à éteindre certains nœuds redondants, sans affecter la couverture maximale du réseau. Les problèmes de connectivité et d'une bonne couverture géo-spatiale sont également abordés dans [140], où les auteurs se préoccupent en même temps de problèmes liés à l'économie d'énergie. Une autre méthode pour prolonger la durée de vie d'un réseau est de sélectionner la route de transmission des données la plus adaptée (i.e. l'itinéraire du paquet). Ainsi, [141] propose une méthode basée sur l'apprentissage d'automates qui choisissent l'itinéraire en fonction d'un paramètre principal, comme le niveau de la consommation d'énergie.

Ainsi, nous pouvons constater que la nécessité d'économiser l'énergie dans les réseaux de capteurs est toujours présente, même si le but principal est d'améliorer d'autres paramètres du réseau. Dans la suite de ce chapitre, nous allons présenter une architecture intégrant des antennes directives, qui tire partie des avantages discutés auparavant.

IV.2.3 Propositions pour l'utilisation d'antennes directives

Les premiers travaux liés aux antennes directives qui visent les réseaux de communication sans fil sont associés au protocole WI-FI (Wireless Fidelity). Donc, les premières solutions existantes sont liées aux spécificités de ce standard [115]. Le principe de fonctionnement est simple : chaque nœud est équipé d'une antenne directive composée de N sous antennes et chaque antenne est dirigée vers un secteur de $360/N$ degrés. Le signal à envoyer est alors propagé dans un seul secteur grâce à une des N antennes et l'antenne à utiliser pour émettre le signal est déterminée selon la position (direction) du récepteur.

La couche MAC, à part son rôle de gestionnaire pour l'accès au canal de transmission, est chargée de déterminer dans quelle direction se trouve un nœud voisin et quelle antenne doit être utilisée pour l'émission. Pour cela, nous devrons étudier les concepts qui ont été utilisés au niveau MAC pour supporter ce nouveau type d'antennes.

Pour gérer l'accès au canal et pour éviter des problèmes comme celui des nœuds cachés, le 802.11 introduit la paire de paquets RTS/CTS (Request to Send / Clear to Send). Aux cours du temps, il a été proposé différents schémas pour échanger ces paquets spécifiques, également en présence d'antennes directives. Les avantages et les inconvénients sont tout d'abord mis en exergue, puis chaque protocole sera présenté individuellement.

Ainsi, nous retrouvons le mode de transmission *Omni-directional RTS/CTS*, qui s'impose avec deux grands inconvénients : il réduit l'étendue des communications et la réutilisation spatiale [116], [117]. En contrepartie, le mode de transmission *Directional RTS/CTS* permet un rayon de communication plus long, avec exploitation de la diversité spatiale, mais il souffre du problème de famine [117].

Une autre solution trouvée dans la littérature est le mode *Multiple-Directional RTS/CTS* qui consiste à transmettre plusieurs fois des paires de paquet, une dans chaque direction, de telle sorte que tous les voisins de l'émetteur et du récepteur soient informés de la communication en cours. Cette méthode évite le problème de famine, mais il a un

dépassement (overhead) important dû aux paquets RTS/CTS multiples. Le protocole qui utilise ce principe est présenté dans [116] et s'appelle Circular-DMac.

Le protocole *DVCS (Directional Virtual Carrier Sensing)* [117] est une adaptation du Virtual Carrier Sensing du 802.11 pour supporter les antennes directives. Cette version permet au protocole MAC de déterminer la disponibilité du canal dans une direction spécifique. Ce concept est implémenté en utilisant une table *DNAV (Directionnal Network Allocation Vector)* qui est une adaptation du NAV utilisé dans le 802.11 de base. Chaque entrée de la table DNAV est associée à une direction, et détermine l'état du canal dans cette dernière. La mise à jour de la table DNAV est importante pour éviter le problème de nœuds cachés, et assurer la diversité spatiale (problèmes existants pour les modes présentés précédemment).

Les protocoles MAC basés sur les antennes directives utilisent « *le cache d'angle d'arrivée* », paramètre essentiel pour déterminer la direction d'un nœud voisin. Autrement dit, quand un nœud reçoit un signal de ses voisins, il va enregistrer l'angle d'arrivée du signal dans son cache. Quand un nœud a un paquet à transmettre, il est envoyé en mode directionnel, à la condition que l'angle d'arrivée du voisin ait été enregistré dans le cache.

Dans notre proposition nous visons entre autres d'améliorer un certain nombre d'éléments qui sont critiques dans un WSN et qui, aujourd'hui, ne sont pas forcément résolus :

- ❖ La précision de synchronisation pour la lecture des nœuds actifs (les capteurs) dans une très petite fenêtre temporelle, qui peut correspondre à quelques dizaines de nanosecondes (aspect qui a été traité dans le Chapitre III).
- ❖ Une bonne connectivité entre les éléments du réseau afin de ne pas affecter la précision de synchronisation des horloges et la qualité de service, par l'intégration des antennes directives.
- ❖ La consommation d'énergie qui est cruciale pour la durée de vie du système car les éléments du réseau sont alimentés par leur propre batterie.

Ainsi, nous allons intégrer dans notre architecture le concept d'antennes directionnelles qui améliore le fonctionnement du réseau, car ces antennes peuvent s'orienter vers un destinataire prévu (Figure IV.6). Au contraire, en utilisant des antennes omnidirectionnelles, tous les signaux inutiles reçus vont être équivalents à des sources de bruit.

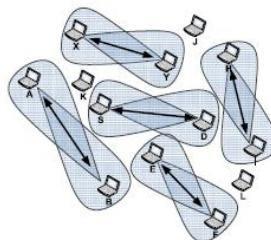


Figure IV.6 Exemple de WSN avec des antennes directionnelles [118].

Les antennes directives sont également utiles pour augmenter la distance de transmission (en augmentant la puissance du signal de transmission / réception), tout en positionnant le lobe principal dans la direction souhaitée.

De plus, notre solution est capable de réaliser d'importantes économies d'énergie en se basant sur le principe de contrôle de la puissance, ce qui va être expliqué dans la suite de ce chapitre.

IV.2.3.1 Solutions existantes pour la couche MAC

Les propositions de protocole MAC les plus importantes basées sur les antennes directionnelles qui utilisent les mécanismes présentés auparavant sont présentés dans cette partie.

LE PROTOCOLE DMAC

Proposé par Choudhury et al [119], DMAC est un protocole qui utilise les paquets de type RTS/CTS, DATA et ACK pour accomplir un échange de données, où la paire RTS/CTS est transmise directionnellement. Après une phase de PCS (Physical Carrier Sensing), DMAC utilise le mécanisme DVCS pour déterminer la disponibilité du canal, avant de commencer la phase de transmission. Quand la couche MAC conclura que le canal est libre pour une émission dans la direction désirée, elle entre dans une phase de « backoff » après laquelle la transmission du paquet RTS peut commencer.

A la réception d'un paquet RTS, la couche Mac détermine la direction d'arrivée du paquet, et envoie un paquet CTS dans la direction inverse. Après l'échange de la paire RTS/CTS les paquets DATA et ACK sont envoyés de manière directive.

Les principaux avantages de ce protocole par rapport au standard 802.11 sont en termes de débit, de délai de transmission, et de diversité spatiale. Mais, en contre partie, il souffre des problèmes de nœuds cachés et de famine et il se présente inadapté aux topologies linéaires (nœuds alignés).

LE PROTOCOLE MMAC

Une autre proposition, le protocole MMAC [115], représente une amélioration de DMAC, dans le but d'accomplir de meilleures performances. Il utilise des paquets « Multi Hop RTS » pour établir des liens directionnel-directionnels. Le principe de fonctionnement est schématisé dans la Figure IV.7, où le nœud A essaye d'établir un lien directionnel-directionnel avec le nœud F.

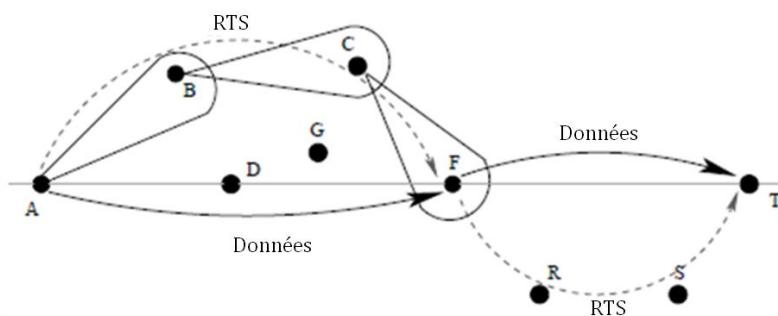


Figure IV.7 Le mécanisme de fonctionnement du MMAC protocole.

L'intérêt d'établir ce type de liens est d'étendre le rayon de communication entre deux nœuds. Dans ce cas, la distance à laquelle deux nœuds peuvent communiquer est augmentée.

L'avantage par rapport au DMAC est que le protocole MMAC étend la distance à laquelle deux nœuds peuvent communiquer. Mais bien entendu, il y a aussi des inconvénients : c'est la plus faible probabilité que le Multi Hop paquet arrive à destination, auquel se rajoute le dépassement dû aux paquets « Forwarding RTS ».

LE PROTOCOLE DMAC CIRCULAIRE

Une autre proposition notable est celle de Korakis et al. [116], qui ont proposé un protocole MAC qui exploite les avantages des antennes directives tout en évitant les problèmes de nœuds cachés et de famine : c'est le DMAC Circulaire. Ce protocole est basé sur la transmission de paquets RTS d'une manière circulaire, cela veut dire sur toutes les antennes pour couvrir toute la zone autour de l'émetteur.

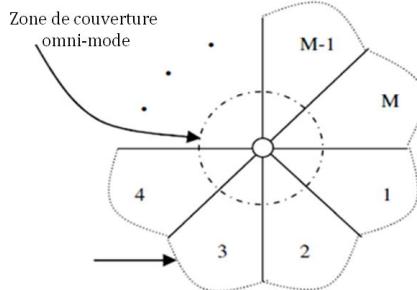


Figure IV.8 Le protocole DMAC Circulaire.

A travers ce mécanisme circulaire, un nœud voisin est découvert quelque soit sa position. Une fois le paquet RTS reçu, le récepteur détermine la direction d'arrivée du RTS et envoie un paquet CTS dans la direction inverse.

Le DMAC circulaire évite les problèmes de nœuds cachés et de famine, mais, en contre partie, le dépassement des paquets RTS circulaire dégrade ses performances.

Mise à part les solutions déjà présentées ci-dessus, nous retrouvons aussi dans la littérature des propositions pour l'amélioration du fonctionnement des différents types de couches MAC, toujours en présence d'antennes directives. Ainsi, dans [120], les auteurs proposent une interface radio IR-UWB reconfigurable avec des antennes directives, qui a la capacité d'assurer aussi la localisation. Leurs travaux sont basés sur la reconfiguration des paramètres (comme le nombre de slots par trame, la durée d'une trame, la durée d'un slot, etc.) au niveau de la sortie de la couche MAC et de l'entrée au niveau PHY, en utilisant la technologie TH-IR-UWB (Time Hopping- Impulse Radio-Ultra Wide Band).

Dans un autre contexte plus critique, de réseaux ad-hoc véhiculaires [121], les auteurs de l'article proposent un modèle de niveau MAC pour faire de la localisation dans des conditions de mobilité. Les informations sur l'emplacement des nœuds mobiles sont ainsi utilisées pour fournir les liens de communication, en utilisant des faisceaux directionnels. Ils montrent aussi que, par l'exploitation de la directivité des antennes, nous pouvons améliorer la fiabilité des liens de communication, même lorsque les directions et les vitesses de déplacement des véhicules ont été modifiées.

Toujours dans le cadre des réseaux ad-hoc véhiculaires, [122] présente la conception et la mise en œuvre d'un protocole MAC multi-canal pour des réseaux très dense, en utilisant le principe de fonctionnement des antennes directionnelles avec un diagramme local de rayonnement d'antennes.

Nous pouvons donc conclure que, l'utilisation d'antennes directionnelles peut réduire considérablement les interférences radio, et se traduit par une meilleure utilisation du support sans fil. Dans les WSNs, les antennes directives peuvent augmenter la réutilisation spatiale du canal et leur gain plus élevé permet d'avoir moins de sauts jusqu'à la destination.

Par conséquent, en tirant avantage des solutions existantes dans la littérature (par exemple le DMAC Circulaire) nous allons proposer un modèle d'antennes directives qui peut être adéquat pour le contexte de réseaux de capteurs sans fil.

IV.2.3.2 *Les principaux protocoles de routage*

Hormis les collaborations inter-couches, il est nécessaire d'étudier quels sont les protocoles de routage qui s'adaptent le mieux à l'utilisation d'antennes directives. Le routage doit assurer la connectivité d'un réseau quand les nœuds radio ne sont pas à portée directe. Un paquet peut donc être relayé par des nœuds intermédiaires pour atteindre sa destination finale.

Les réseaux ad-hoc représentent historiquement ceux qui ont posé les bases des protocoles de routage pour les réseaux de capteurs, car la problématique d'extension de la zone de couverture par le routage est similaire ; des protocoles plus spécifiques ont ensuite été proposés pour s'adapter aux contraintes des réseaux de capteurs.

Ainsi, en 1997 a été créé un groupe de travail appelé MANET (Mobile Ad Hoc Networks) qui réunis l'ensemble des spécifications des protocoles de routage pour les réseaux sans fil ad hoc. Nous pouvons donc distinguer deux types de protocoles de routage (réactifs et proactifs), présentés dans le Tableau IV.2 :

| Types de protocoles | Caractéristiques | Protocoles existants |
|---------------------|---|---|
| Réactifs | <ul style="list-style-type: none"> - découvre et calcule une route uniquement à la demande d'une application, en utilisant un paquet de requête envoyé en diffusion. - le délai d'attente d'un paquet pour être acheminé vers la destination augmente du fait que l'on ne dispose pas d'une route immédiatement disponible - la solution prévue est d'introduire des files d'attente. - permet de réduire le trafic de contrôle du protocole de routage. - coûteux pour des réseaux denses. | <ul style="list-style-type: none"> -AODV (Ad hoc On-demand Distance Vector Routing)[123]. -DSR (Dynamic Source Routing Protocol) [124]. -TORA (Temporally Ordered Routing Algorithm) [125]. -ABR (Associativity Based Routing) [126] -SSR (Signal Stability Routing) [127] |
| Proactifs | <ul style="list-style-type: none"> - chaque élément du réseau détient une/plusieurs tables de routage permettant d'atteindre tous les nœuds du réseau. - la mise à jour des routes se réalise par l'échange de trames périodiques de contrôle. - échange abusif d'informations pour des tables de routage de faible taille dans un réseau fluide. | <ul style="list-style-type: none"> -DSDV (Destination Sequenced Distance-Vector) [128] -WRP (Wireless Routing Protocol) [129] -OLSR (Optmised Link State Routing) [130] -FSR (Fisheye State Routing) [131] |

Tableau IV.2 Comparaison des protocoles réactifs et protocoles proactifs.

La communauté académique a proposé également des solutions de routage hybrides (c'est à dire des protocoles proactifs en zone locale et réactives entre zones), ainsi que des techniques de routage géographique. Dans [132], les auteurs proposent une stratégie de routage pour des réseaux maillés. L'idée principale est de déployer le réseau en utilisant un certain type de diagramme géométrique et ensuite de concevoir un protocole de routage géographique qui exploite les propriétés de routage de ce graphe.

Nous allons maintenant présenter les mécanismes pour les autres types de protocoles de routages que ceux classifiés ci-dessus, dans le but de couvrir des catégories de réseaux de communications sans fil plus spécifiques.

LES PROTOCOLES HIERARCHIQUES

Pour ces protocoles, le réseau est divisé en plusieurs groupes de nœuds (ou clusters) avec un chef pour chaque cluster. Le chef (cluster head) doit transmettre les messages diffusés dans son cluster aux autres chefs de groupes, pour les diriger à la destination finale. Le choix du chef se fait selon une politique pré définie (comme le nœud avec le plus grand nombre de voisins, selon le niveau de l'énergie résiduelle ou bien à tour de rôle, etc.).

Dans cette catégorie, nous pouvons citer le protocole LEACH (Low-Energy Adaptive Clustering Hierarchy) [133] qui gère l'élection périodique du cluster head en fonction du pourcentage de nombre de clusters et du nombre de tour depuis lequel le nœud n'a pas été élu. En moyenne, 5% des nœuds sont choisis pour devenir chefs de groupes.

L'objectif de TEEN (Threshold sensitive Energy Efficient sensor Network protocol) [134] est de transmettre uniquement quand la valeur détectée est supérieure à un certain seuil fixé par le cluster head. Ensuite, il ne transmet pas le même message tant que sa valeur n'a pas changé d'un seuil de variation (aussi fixé par le cluster head).

LES PROTOCOLES CENTRES SUR LES DONNEES

Ces protocoles sont utilisés pour des réseaux dans lesquels il est difficile d'avoir des identifiants (adresses) de niveau MAC ou réseau. Ceci est d'autant plus vrai que les nœuds sont simplistes et s'appliquent plus particulièrement aux réseaux de capteurs, car ils ne demandent pas de mécanisme d'adressage. Les informations sont ainsi propagées de proche en proche. Le principe consiste à annoncer tout d'abord les données à envoyer, les nœuds intéressés répondront par un ACK et ensuite les données seront envoyées.

Il existe deux mécanismes de diffusion, qui ne demandent pas une connaissance de la topologie :

- ❖ **Le flooding** – consiste à diffuser un message à tous les éléments du réseau, qui se propage pour atteindre la destination ou jusqu'à un nombre maximum de sauts.
- ❖ **Le gossiping** - consiste à choisir aléatoirement un voisin pour lui faire passer le message.

Nous pouvons citer ici le protocole SPIN (Sensor Protocol for Information via Negotiation) [135] qui envoie un message d'annonce avant d'envoyer les données ; les voisins intéressés demandent aussi la réception des données, qui vont être ensuite retransmises.

Une autre méthode est la diffusion directe (Directed Diffusion) [136] des données. Le principe consiste à envoyer une requête qui va être prise en compte par les voisins, qui à leurs tour vont rediffuser la requête. Ceci suppose une connaissance géographique du

réseau pour orienter les requêtes dans les régions où la donnée recherchée est censée être produite.

LES PROTOCOLES BASES SUR LA POSITION

Dans un réseau de capteurs, pouvoir localiser un nœud est plus important que connaître son identifiant (l'adresse). Ainsi, ces protocoles prennent en compte l'hypothèse que chaque nœud connaît sa position et qu'il est capable de connaître aussi la position des autres éléments du réseau.

Dans cette catégorie de protocoles de routage se trouve GEAR (Geographically and Energy Aware Routing) [137] qui fonctionne sur le principe de découpage du réseau en plusieurs régions. Un autre aspect important est que chaque nœud connaît le coût pour atteindre chacune des régions ainsi formées.

Dans la phase d'acheminement, le paquet contient l'information de la région destination et chaque nœud connaît la position et l'énergie résiduelle de lui-même et de ses voisins (à la demande). Le protocole permet aussi l'ouverture d'un lien direct entre deux nœuds qui sont à portée et avec un niveau d'énergie qui leur permet d'effectuer l'envoi.

Cet état de l'art (section IV.2.3.1 et IV.2.3.2) couvre les contributions les plus significatives, récentes ou plus anciennes, concernant les protocoles d'accès au medium et de routages pour les réseaux de communication sans fil. Les protocoles de niveau MAC ainsi présentés permettent la prise en compte des systèmes de communication reconfigurables, tels que ceux visés par notre application. L'observation générale est que la plupart des travaux utilisant la reconfiguration, mettent en avant la reconfiguration de la puissance d'émission pour faire du contrôle de la puissance [138].

IV.3 L'applicabilité dans une catégorie large de réseaux sans fil

Nous nous proposons dans cette section de présenter (d'une manière générale et suivie après par un exemple concret de configuration de réseau) les catégories de systèmes de communication qui puissent supporter un modèle d'antennes directives. Les résultats de simulation ainsi obtenus, vont être présentés et analysés dans le Chapitre V (Simulation et expérimentations).

L'APPORT DES ANTENNES DIRECTIVES DANS UNE ARCHITECTURE FIXE DE RESEAU

Pour expliquer l'intégration d'antennes directives dans notre architecture, nous présentons l'exemple de l'application SACER où les nœuds capteurs sont fixes.

Notre réseau de référence (Chapitre II) est composé de 73 éléments par aile, qui communiquent en vol. L'élément central (C) est conçu pour recueillir toutes les informations saisies par les nœuds/capteurs à travers les routeurs, et de les envoyer à l'intérieur de la cabine ; à leur tour, chacun des routeurs (R) doit recueillir les informations auprès de ses nœuds associés (Ns) (Figure IV.9).

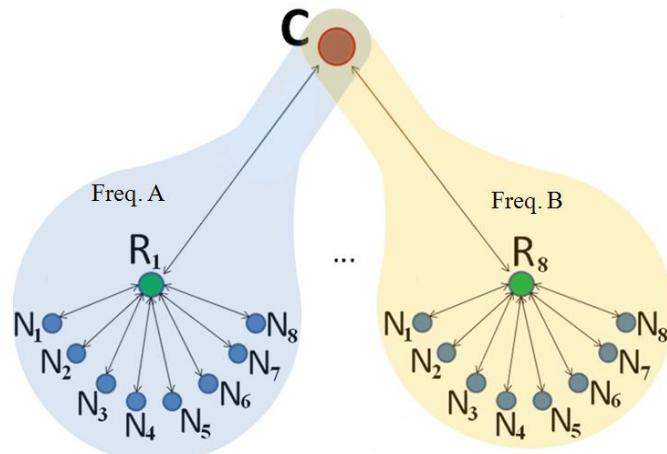


Figure IV.9 L'organisation hiérarchique du réseau.

Donc, nous avons un système de communication composé de huit sous réseaux, dans lequel le concentrateur est conçu selon la méthode des interfaces multiples où chaque sous-réseau utilise une fréquence différente (fréquence A, fréquence B, etc.). Les communications dans chaque sous réseau sont organisées par un mécanisme d'accès au médium de type TDMA fixe, où le temps de parole de chaque élément est prédéterminé. L'avantage de ce type d'organisation est l'évitement de possibles interférences et la réduction des délais de transmission.

Dans cette configuration du réseau, notre modèle d'antennes directionnelles va être évalué, ce qui signifie que chaque élément du réseau est équipé d'un nouveau mode de transmission.

Le trafic véhiculé dans le réseau pour analyser le comportement du système en présence du modèle proposé, correspond à notre protocole de synchronisation 1588-PBS hybride (section III.3.1). Le mode de fonctionnement de ce protocole de synchronisation est basé sur l'idée qu'un groupe de nœuds (les nœuds X) se synchronisent en entendant les messages de synchronisation d'un couple maître-esclave (M-S), en utilisant l'algorithme de fonctionnement du protocole PBS. A leur tour, les maîtres et les nœuds esclaves se synchronisent en utilisant le standard IEEE-1588 (Figure III.6).

Notre architecture de réseau décrite ci-dessus (Figure IV.9), comporte deux niveaux de synchronisation: le concentrateur-routeurs (Niveau 1) et le routeurs-nœuds (Niveau 2).

Le premier niveau utilise l'algorithme du standard IEEE1588, correspondant aux messages échangés entre le couple M-S.

Pour le deuxième niveau de synchronisation, si nous considérons le sous-réseau qui utilise la fréquence de transmission A, nous avons un routeur et un nœud qui forme un couple M-S, tandis que les sept autres nœuds seront identifiés avec le X-nœud.

IV.4 Principe et implémentation de notre solution

Nous proposons ici un modèle d'antennes directionnelles pour les réseaux d'instrumentations sans fil que nous utilisons par la suite dans la simulation de notre architecture. Notre modèle peut être appliqué à un large éventail d'applications, mais ici nous démontrons la faisabilité et le principe de fonctionnement pour le domaine de l'instrumentation sans fil et plus particulièrement pour notre application.

Nous allons ainsi nous concentrer sur le développement d'une nouvelle architecture de communication qui permette d'optimiser le rendement du réseau en terme de débit et de délai, de diminuer les collisions et de réaliser une réduction de la consommation d'énergie.

La stratégie de développement est basée sur l'intégration des antennes directives dites intelligentes dans l'architecture physique de communication.

Les recherches sur les réseaux sans fil de type WI-FI (ayant comme support de transmission des antennes directives) présentés par avant, montrent leurs performances et leur efficacité. Notre travail inspiré de ces études consiste à étudier l'effet de l'utilisation de ces antennes pour la couche MAC-TDMA sans fil, de concevoir une architecture Cross-Layer pour les réseaux d'instrumentations, et de découvrir et analyser les avantages et les problèmes qui en découlent. De plus, nous allons étudier le comportement de notre modèle en présence du trafic de synchronisation analysé dans le chapitre précédent.

La première partie de la phase d'implémentation consiste en la modélisation des antennes directives dans NS-2 ; cette étape est suivie par leur intégration dans le modèle MAC-TDMA déjà existant. Le nouveau modèle ainsi conçu est nommé le *Directionnal MAC - TDMA*.

IV.4.1 Proposition d'architecture d'antennes directives

Précédemment, nous avons conclu qu'un moyen d'améliorer la qualité de service dans notre réseau est de profiter des avantages offerts par les antennes directionnelles. Ainsi, notre proposition est fondée sur l'intégration de ce type d'antennes dans l'architecture physique de la couche MAC-TDMA.

Dans notre architecture, chaque nœud est équipé de quatre antennes directionnelles et d'une cinquième isotrope. Le choix pour cette configuration (4 antennes directives et une antenne omnidirectionnelle) a été fait en prenant en compte des paramètres comme la taille de l'antenne qui représente $\frac{1}{2}$ de la longueur d'onde et qui est influencée par la fréquence utilisée (plus la fréquence augmente, plus la taille de l'antenne diminue), ou le coût de fabrication.

Chaque antenne directionnelle a la capacité de transmettre et recevoir des données, et se caractérise par son gain de transmission et de réception (G_t , G_r) dans une direction spécifique (Figure IV.10).

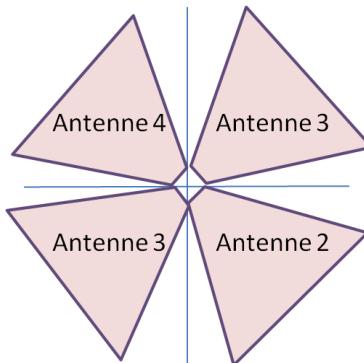


Figure IV.10 Notre architecture d'antennes directionnelles.

L'antenne isotrope présente un gain de 1 dB uniforme dans toutes les directions, et elle est utilisée uniquement en mode de transmission broadcast.

Notre nouvelle couche physique est chargée de transmettre et recevoir des trames sur le canal radio. A la couche standard de base, les rôles suivants ont été ajoutés:

l'activation / désactivation des antennes, la détermination de l'angle d'arrivée du signal et la détermination de la puissance du signal reçu. L'expression de la puissance de réception est la suivante :

$$P_r = \frac{P_t \times G_t \times G_r \times \lambda^2 \times d_{ref}^2}{4(\Pi \times d)^2} \quad (\text{IV.1})$$

où P_t et P_r sont respectivement les puissances de transmission et de réception, G_t et G_r sont les gains pour la transmission et pour la réception, λ représente la longueur d'onde, d est la distance entre les nœuds et d_{ref} est la distance de référence donnée par l'expression $d_{ref} = 2D/\lambda^2$ (D représente la taille maximale de l'antenne de référence).

Dans notre étude, nous avons considéré que l'antenne de référence est une antenne dipôle demi-onde, où $D = (\lambda/2)^2$. Dans ces conditions, l'expression de la puissance de réception devient :

$$P_r = \frac{P_t \times G_t \times G_r \times \lambda^2}{4(\Pi \times d)^2} \quad (\text{IV. 2})$$

Ensuite, nous calculons la puissance d'émission nécessaire pour assurer la connectivité entre les éléments du réseau dans la configuration de notre modèle d'antenne. Pour cela, nous devons calculer la distance maximale à laquelle une antenne omnidirectionnelle peut émettre / recevoir (équation IV.3).

$$P_r = \frac{P_{t_MAX_OMNI} \times \lambda^2}{(4\Pi \times d_{MAX})^2} \quad (\text{IV. 3})$$

Nous avons également calculé la distance maximale à laquelle une antenne directionnelle peut émettre / recevoir (équation IV. 4).

$$P_r = \frac{P_{t_MAX_DIR} \times G_t \times G_r \times \lambda^2}{(4\Pi \times d_{MAX})^2} \quad (\text{IV. 4})$$

Nous obtenons finalement que la puissance de transmission pour une antenne directionnelle (par rapport à celle d'une antenne omnidirectionnelle) est la suivante:

$$P_{t_MAX_DIR} = \frac{P_{t_MAX_OMNI}}{G_t \times G_r} \quad (\text{IV. 5})$$

L'équation IV.5 montre que l'utilisation d'antennes directionnelles permet la réduction de la puissance de transmission / réception ($G_t \times G_r$ fois plus petite), donc une importante économie d'énergie peut être réalisée.

Un autre aspect important à considérer est lié aux méthodes de conception et aux coûts de fabrication pour une telle antenne.

En raison de l'évolution de la Micro-électronique (MEMS) [142], aujourd'hui nous pouvons produire des antennes à faisceaux commutés (switched antenna) avec une haute performance en termes de gain (mieux que l'antenne purement électroniques, comme les antennes imprimées [143] [144]). L'avantage de cette technologie est que l'antenne ne doit pas être orientée mécaniquement, il faut juste que la bonne antenne soit activée.

Une fois établis les détails du modèle, l'étape suivante est d'implémenter ces antennes directionnelles dans un langage de simulation comme NS-2 et de l'intégrer dans le MAC-TDMA déjà existant.

IV.4.2 L'implémentation de notre solution

Pour tester le comportement et les performances de l'architecture Hybride 1588-PBS Directif, nous avons utilisé et étendu le simulateur NS-2.

Le simulateur NS-2 offre un modèle de couche physique général, utilisé par tous les protocoles pour les réseaux sans fil, basé sur les antennes omnidirectionnelles. Nous avons donc besoin d'adapter la fonctionnalité de ce niveau à notre modèle d'antennes. Par conséquent, nous avons fait des changements pour le mode de transmission et de réception. Le niveau physique remplit également la fonction de synchronisation des horloges pour tous les éléments du réseau (Figure IV.11).

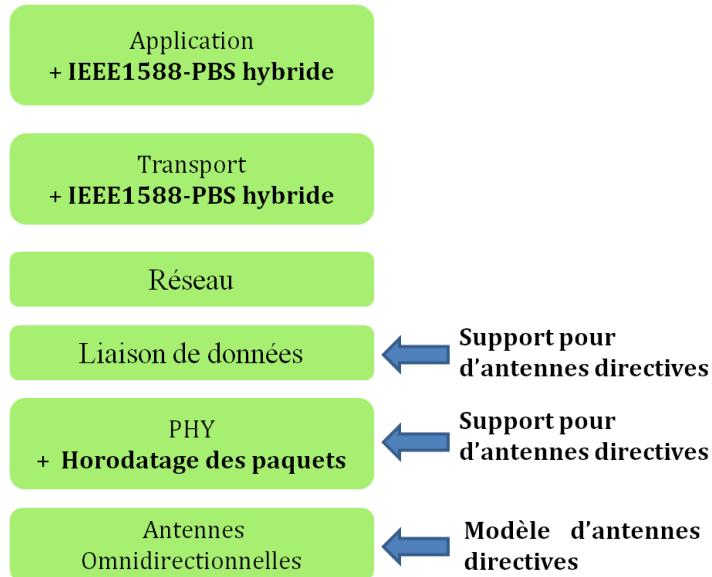


Figure IV.11 La pile de protocoles utilisés pour la synchronisation 1588-PBS hybride.

Ce schéma présenté ci-dessus (Figure IV.11) correspond à l'architecture du protocole de synchronisation 1588-PBS hybride. La couche MAC-TDMA, gère les nouvelles fonctionnalités suivantes:

- ❖ La commutation entre les cinq antennes (les antennes directionnelles et l'antenne isotrope) selon la position du nœud destination et le type de trafic (broadcast ou autre type de trafic);
- ❖ L'enregistrement de la position des nœuds voisins ou de l'angle d'arrivée (AOA) des paquets, ce qui est essentiel pour choisir l'antenne de transmission appropriée;
- ❖ La mobilité des éléments

Nous allons maintenant expliquer le mécanisme de transmission et de réception qui assure la connectivité:

- ❖ Pour chaque paquet transmis vers le nœud x, la couche MAC contrôle si l'AOA a déjà été enregistré pour le nœud X. Si l'angle d'arrivée est présent, la couche MAC informe la couche Physique pour effectuer l'activation de l'antenne correspondante et, en conséquence, pour la transmission de paquet. Sinon, la couche MAC informe la couche Physique d'activer l'antenne isotrope et de désactiver toutes les autres antennes (le cas de transmission en broadcast).

- ❖ Pour chaque paquet reçu, la couche MAC enregistre dans un champ les informations [l'adresse source, l'antenne de réception], ce qui est important car cela permet au niveau MAC de choisir directement l'antenne d'émission correspondante.
- ❖ Les messages qui permettent la mise à jour de la table sont notamment ceux qui sont envoyés en mode diffusion (broadcast) (c'est-à-dire le *sync_message*, *le follow_up*, *le delay_request* (Figure III.6)).
- ❖ La nouvelle couche MAC est capable de détecter les changements dans la position d'un nœud voisin (à savoir la nouvelle orientation dans l'espace). Elle intègre une fonction qui est chargée de mettre à jour le champ qui contient les directions des voisins (le champ est validé ou pas), en envoyant régulièrement des messages spécifiques uniquement en mode broadcast à travers l'antenne isotrope. Pour éviter la surcharge inutile du trafic, nous avons intégré ce service dans des messages de contrôle spécifiques du réseau.

IV.5 Conclusion

Dans l'optique d'améliorer les performances du système, nous avons proposé un modèle d'antennes directionnelles qui, à part les avantages d'une bonne couverture, d'une bonne connectivité, d'une réduction des interférences et des retards, etc., manifeste la capacité de prolonger la durée de vie du système.

Notre travail a été motivé par les études existantes ainsi que par notre application réelle, qui consiste à étudier les effets de l'utilisation de ces antennes dans la méthode TDMA, pour la conception d'une architecture inter-couches dédiée aux réseaux de capteurs. Cette phase de simulation est une condition préalable pour la réalisation d'un prototype, et pour la mise en œuvre de nos propositions dans le cadre d'un prototype réel.

Il a été discuté précédemment et il va être montré dans le chapitre des simulations que notre modèle d'antenne directionnelle est conçu de façon à être efficace en termes de *couverte*, de *connectivité*, d'*économies d'énergie* et même en termes de *coûts de fabrication*.

Dans ce chapitre, nous avons proposé un modèle d'antennes directionnelle que nous avons intégré dans le contrôle d'accès au medium TDMA. Cela implique également une adaptation pour d'autres protocoles de la pile de communication, à savoir : la couche Physique, Transport, Canal de transmission et le modèle d'Energie. Nous avons intégré dans le simulateur NS-2 la configuration de notre réseau et les modifications nécessaires pour pouvoir tester le nouveau protocole Directionnal MAC -TDMA.

Dans le dernier, chapitre nous allons donc présenter les résultats concernant les propositions pour l'optimisation de notre système, ainsi que les différents tests et mesures qui ont été effectués.

V SIMULATIONS ET EXPERIMENTATIONS

| | | |
|------------|--|------------|
| V.1 | Introduction | 119 |
| V.2 | Topologie : choix et validation | 120 |
| V.2.1 | Objectif : déterminer la meilleure topologie du système | 120 |
| V.2.2 | Intégration des interfaces multiples | 122 |
| V.2.3 | Synthèse | 127 |
| V.3 | La synchronisation vs. l'économie d'énergie | 127 |
| V.3.1 | Evaluation des solutions existantes..... | 128 |
| V.3.2 | Notre proposition de synchronisation. Analyse comparative des performances | 138 |
| V.3.3 | Conclusion | 142 |
| V.4 | La directivité des antennes vs. l'économie d'énergie | 142 |
| V.4.1 | Analyse comparative des performances..... | 142 |
| V.4.2 | Conclusion | 146 |
| V.5 | Synthèse générale | 146 |

Dans ce chapitre, nous allons exposer l'ensemble d'expérimentations (matérielles) et des simulations (réalisées avec le logiciel de simulation de réseaux NS-2) que nous avons réalisées, dans le but de valider : le choix de la topologie réseau du système, la solution proposée pour la synchronisation et le modèle des antennes directives. L'intérêt de cette partie est de montrer le mode de fonctionnement de notre système dans différentes conditions et d'évaluer les solutions proposées pour notre réseau d'instrumentation et de mesures, qui permettront de mieux traiter les contraintes en terme de précision de synchronisation et de gestion d'énergie de notre application.

V.1 Introduction

Le premier objectif de nos travaux a été de proposer l'architecture de communication la mieux adaptée pour un contexte de réseaux de capteurs sans fil. Ainsi, le processus de conception a commencé avec la proposition de la topologie, qui dans notre application SACER est composée de plusieurs niveaux de communication, pour être en mesure d'atteindre les dispositifs les plus éloignés.

Pour résoudre les problèmes le plus contraignants du projet, la précision de la synchronisation (avec les paramètres associés comme le Clock Synchronization Precision, ou le CSP) et la gestion d'énergie, nous avons proposé des solutions qui peuvent être très bien intégrées dans notre contexte applicatif.

Par conséquent, nous devons être capables d'assurer une très bonne couverture et connectivité du réseau, une très bonne précision de la synchronisation, tout en proposant des stratégies pour économiser l'énergie, qui est une spécification primordiale pour

prolonger la durée de vie du système. Ces travaux, ainsi que les résultats associés vont être présentés par la suite de ce chapitre.

V.2 Topologie : choix et validation

Nous notons que l'objet de notre étude est un réseau de capteurs sans fil organisé dans une topologie hiérarchique, comme représenté dans la Figure V.1:

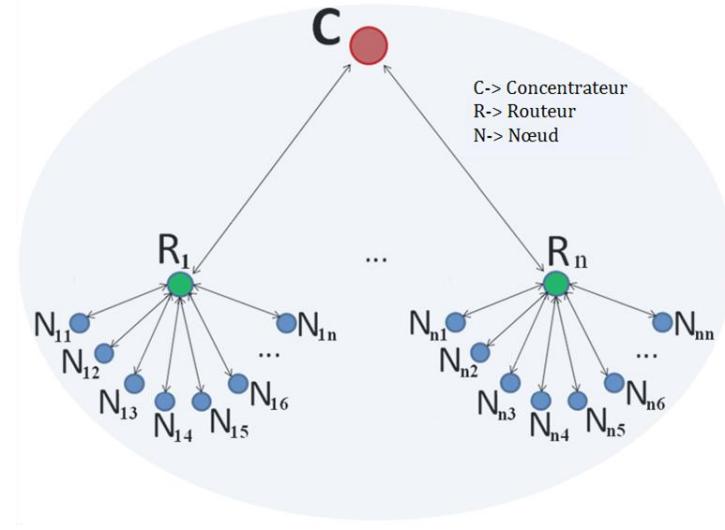


Figure V.1 La topologie réseau de notre système.

Les éléments du réseau (le concentrateur, les routeurs et les nœuds) ont comme rôle principal d'assurer une communication fiable, dans le but de transmettre en mode quasi-synchrone dans l'intérieur de la cabine, les prises de mesures (de pression et de température) réalisées par les nœuds capteurs.

Cependant, le concentrateur reste l'élément le plus complexe (il regroupe toutes les informations capturées chaque seconde par les éléments actifs) et bénéficie d'une alimentation en continu. Les nœuds capteurs sont chargés de la prise des mesures (sans les stocker) et de leurs transmissions, et sont alimentés par batterie. En ce qui concerne les routeurs, leur rôle est de récupérer les informations de la part de nœuds, de les agréger et de les envoyer au concentrateur. Dans ces conditions, pour avoir un coût faible de fabrication (vu le grand nombre des éléments), la complexité de la technologie de production de capteurs et de routeurs doit rester la même ou semblable.

Dans le choix de cette topologie, nous avons justifié auparavant que la présence des routeurs a été obligatoire (ils permettent de réduire les distances de communication, qui implique des avantages comme : une meilleure couverture du réseau, une économie d'énergie du système, etc.). Ainsi, pour trouver le nombre optimal de routeurs nécessaires dans notre système, nous avons analysé à travers des simulations, plusieurs scénarios possibles. Cette analyse fait l'objet de la section ci-dessous.

V.2.1 Objectif : déterminer la meilleure topologie du système

Pour valider le choix en terme de nombre de routeurs nécessaires dans notre réseau, nous avons décidé d'analyser plusieurs scénarios d'organisation du système, à travers le

modèle de simulation décrit dans le chapitre II. Le paramètre que nous allons examiner dans cette analyse sera le temps d'une session complète de communication ou T_{cycle} , (Figure V.2):

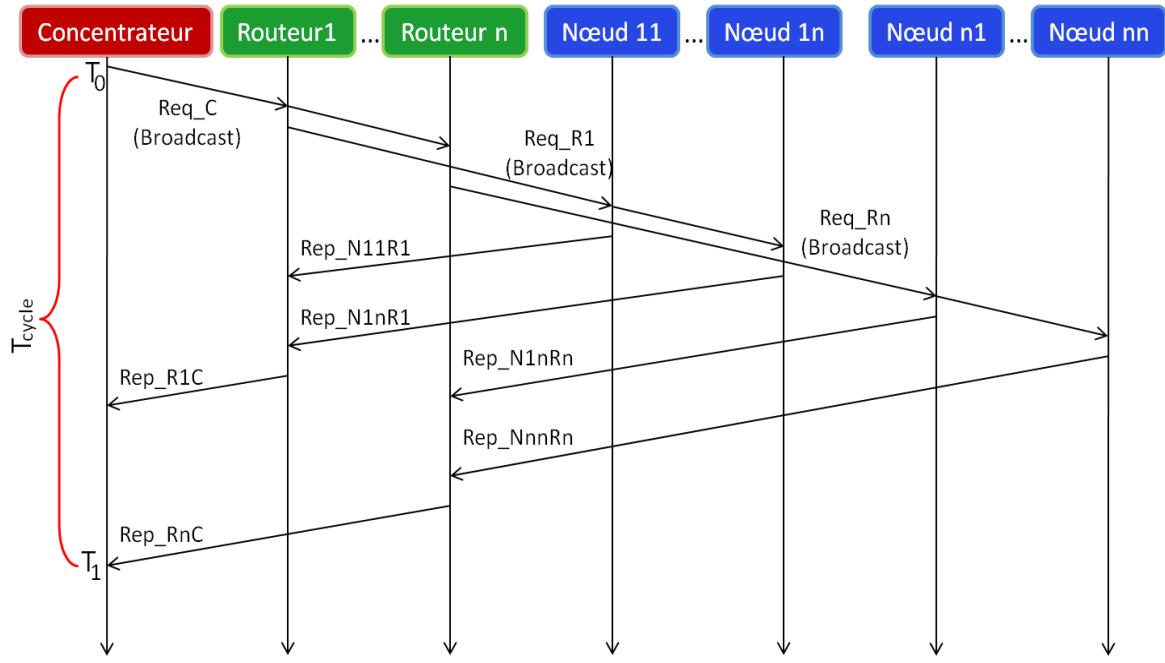


Figure V.2 Le temps d'un cycle de communication (cycle normal de données).

D'après la composition du temps de cycle (Figure V.2), nous proposons par la suite d'effectuer une analyse du comportement et des performances du réseau dans plusieurs scénarios. Le paramètre variable dans toutes ces configurations va être le nombre de routeurs par aile, ce qui nous permet d'obtenir (Figure V.3):

- ❖ Cas n°1: 12 routeurs avec 5 nœuds par routeur ;
- ❖ Cas n°2: 8 routeurs avec 8 nœuds par routeur ;
- ❖ Cas n°3: 4 routeurs avec 17 nœuds par routeur.

Nous notons que pour l'ensemble des cas analysés le nombre d'éléments reste le même (i.e. 73 éléments).

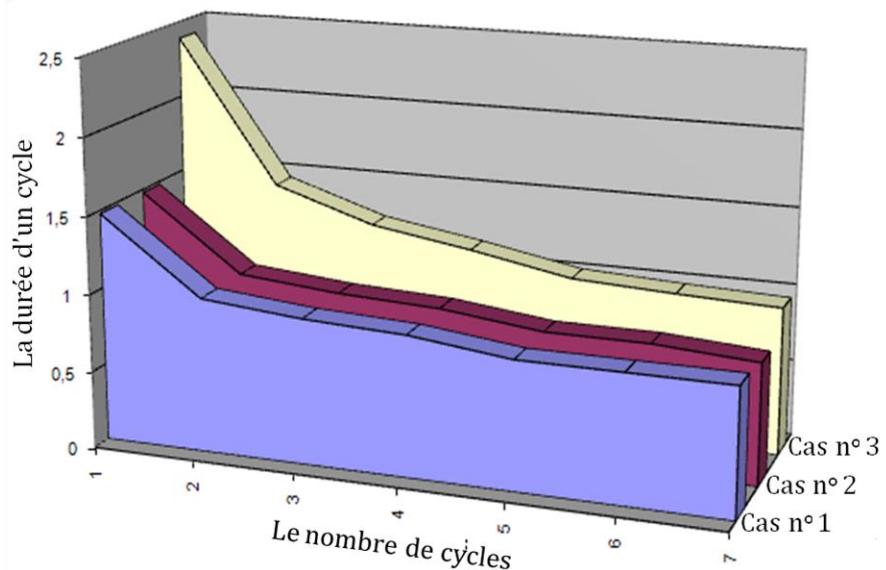


Figure V.3 Comparaison sur les trois cas étudiés.

La première remarque est que, pour tous les cas considérés ci-dessus, nous avons constaté que la durée du premier cycle de communication est plus élevée et qu'elle converge vers une valeur qui reste à peu près constante. Ce phénomène s'explique par le fait que le système se retrouve, à son démarrage, dans une phase d'initialisation et de vérification, phase qui rend le trafic plus chargé.

Nous constatons également que la durée d'un cycle dépend du nombre de routeurs présents dans le réseau. Plus exactement, si le nombre de routeurs est faible (cas n°3), nous enregistrerons une augmentation de la durée du cycle par rapport aux deux autres cas, donc des retards de transmission plus élevés. En ce qui concerne le premier cas (cas n°1), il présente des valeurs pour la durée du cycle légèrement supérieures à celles obtenues dans le cas n°2.

Nous pouvons conclure que le nombre de routeurs a un impact direct sur le temps de propagation en réseau. En même temps, si le nombre de routeurs est trop élevé par rapport au nombre d'éléments actifs (cas n°1), nous perdons l'avantage de l'agrégation des données. Donc, la meilleure organisation proposée pour notre réseau correspond au cas n°2 (soit 8 routeurs par aile).

Pour cela, dans la section suivante nous allons nous concentrer sur la façon de concevoir des routeurs pour qu'ils soient optimaux en termes de performances, avec un coût de production minimal.

V.2.2 Intégration des interfaces multiples

La stratégie adoptée pour la conception de routeur nous conduit à proposer une architecture pour celui-ci, dans le but d'obtenir les meilleures performances du réseau, principalement en termes de précision de la synchronisation, de durée de vie du réseau et de coût de fabrication. Ainsi, nous proposons de concevoir un algorithme capable d'adapter notre système à un mode de fonctionnement basé sur des interfaces de communication multiples.

La configuration de notre système comporte jusqu'à présent: une organisation hiérarchique avec un niveau intermédiaire de communication (les routeurs), et une méthode d'accès au médium de type TDMA, qui permet d'optimiser l'occupation du canal. Par la suite, nous allons nous proposer de mettre en œuvre la méthode d'interfaces multiples pour les éléments du réseau. En ce qui concerne la méthode de routage, nous avons décidé d'utiliser le protocole AODV, qui est l'un des protocoles de routage les plus utilisés dans les réseaux sans fil, car il comporte de nombreux avantages [123].

Le choix pour une méthode de routage différente de celle proposée dès le début pour notre application (le protocole NOAH avec notre propre routage au niveau transport) se justifie par l'intention de placer cette application dans un cadre plus général. Autrement dit, nous souhaitons vérifier le comportement du système équipé avec des interfaces multiples en présence d'un trafic de synchronisation comme proposé dans la section III.3 du chapitre III.

Par conséquent, nous allons tester le comportement de ce réseau en utilisant le protocole de synchronisation IEEE1588-PBS hybride [146]. Nous rappelons ci-dessous le trafic spécifique généré par ce protocole (Figure V.4), où M et S représentent le maître et

l'esclave (correspondant dans le réseau aux groupes concentrateur-routeurs par exemple) et X représente un esclave qui ne fait que recevoir des messages de synchronisation (correspondant à des nœuds spécifiques).

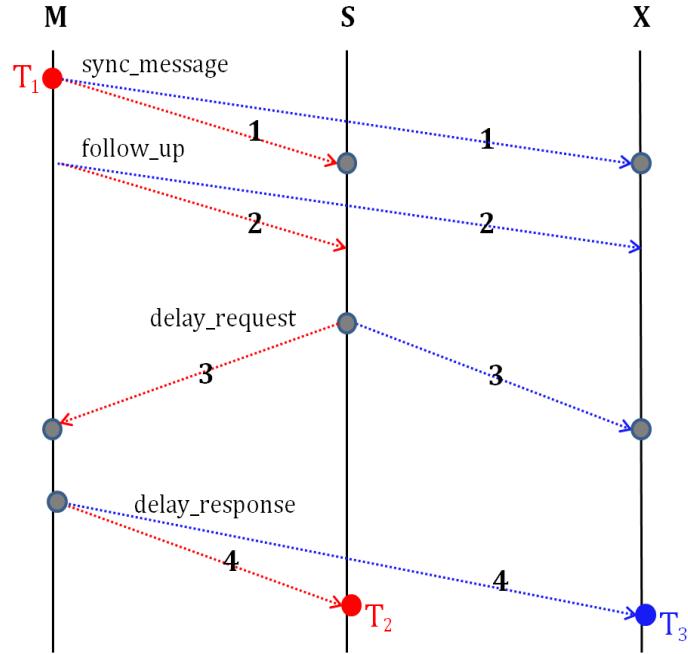


Figure V.4 L'échange des messages dans le protocole IEEE1588-PBS hybride.

Nous calculons ainsi le cycle de synchronisation (selon le type du nœud) avec les relations suivantes:

$$T_{\text{Cycle_MS}} = T_2 - T_1 \quad (\text{V.1})$$

$$T_{\text{Cycle_MX}} = T_3 - T_1 \quad (\text{V.2})$$

Dans cette configuration, nos objectifs sont:

- ❖ De vérifier que la précision de la synchronisation n'est pas affectée par rapport aux performances (des performances pour la précision de la synchronisation comprises entre quelques dizaines de microsecondes et quelques centaines de nanosecondes) obtenues dans [146];
- ❖ D'analyser les retards de transmission (obtenues à partir de l'analyse comparative des paramètres des équations V.1 et V.2);
- ❖ D'analyser la consommation d'énergie aux niveaux des routeurs et des nœuds (le niveau concentrateur nous intéresse moins au niveau de consommation car il est alimenté en continu).

Nous réalisons ce type d'analyse pour deux scénarios différents (section II.4.2.2), que nous les rappelons dans la Figure V.5.

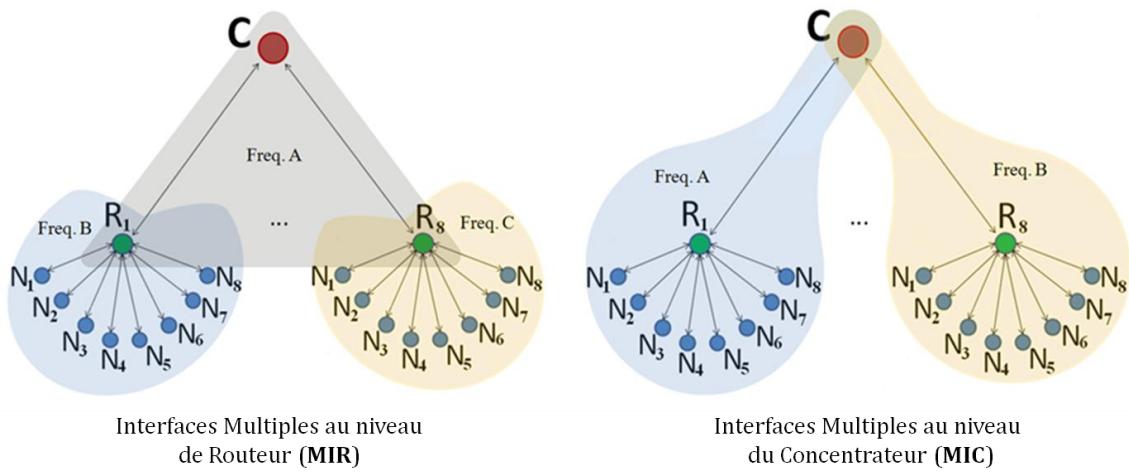


Figure V.5 Deux scénarios d'organisations du réseau avec des interfaces multiples.

Le but est de montrer à travers une analyse comparative les avantages portés par l'organisation que nous avons proposée (l'organisation MIC du réseau), en termes de synchronisation et d'économie d'énergie.

LA PRECISION DE LA SYNCHRONISATION

Tout d'abord, nous allons vérifier si les performances en terme de précision de la synchronisation pour le scénario de type MIC corresponds à nos attentes (autour de la nano seconde).

Nous avons effectué les simulations sur le modèle NS-2 présenté auparavant, pour un ensemble de 73 éléments (un concentrateur, 8 routeurs et 64 nœuds). Grâce à l'organisation hiérarchique du réseau, nous avons constaté que le comportement d'un sous-réseau composé d'un concentrateur (C), un routeur (R) et huit nœuds (N) est similaire à celui des autres sous-réseaux. Pour cette raison et pour avoir des figures plus claires, nous avons décidé de présenter des résultats concernant uniquement un de ces sous-réseaux.

Ainsi, dans la Figure V.6 (a), nous pouvons remarquer que la précision de la synchronisation pour le premier niveau Concentrateur-Routeurs correspond aux performances d'organisation de type MIR en présence du protocole IEEE1588-PBS hybride [146], car les valeurs évoluent entre 10 ns et 230 ns.

En ce qui concerne le deuxième niveau, Routeurs-Nœuds, nous constatons (Figure V.6 (b)) que la valeur moyenne pour la précision de la synchronisation est de l'ordre de quelques centaines de nanosecondes, ce qui correspond aux résultats montré dans [146] pour une organisation de type MIR.

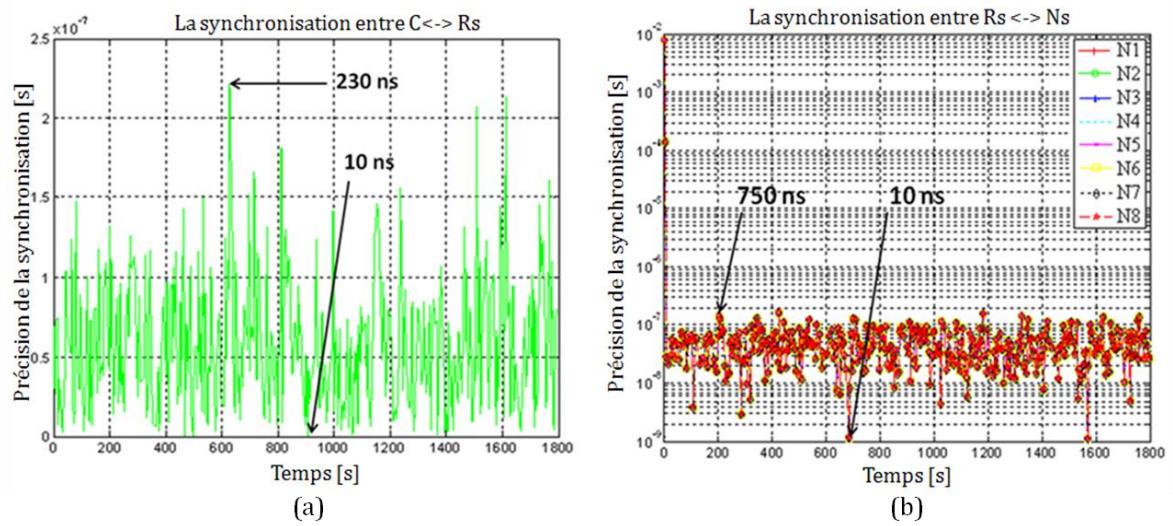


Figure V.6 La synchronisation pour le Niveau1 et le Niveau2 de communication

Nous pouvons donc conclure, en se basant sur les résultats présentés ci-dessus, que notre solution (l'organisation de type MIC) préserve les performances en terme de précision pour la synchronisation du protocole IEEE1588-PBS hybride.

LES DELAIS DE TRANSMISSION

Nous continuons notre série de tests avec l'analyse d'un autre paramètre important, le délai de transmission. Plus précisément, nous nous intéressons à une analyse comparative entre les deux scénarios proposés en début de cette section, pour lesquelles nous mesurons la valeur correspondante à un cycle de synchronisation (équations V.1 et V.2), pour les deux niveaux (respectivement Concentrateur-Routeurs et Routeur-Nœuds).

Dans les figures ci-dessus, nous représentons la durée d'un cycle complet de communication réalisé entre le concentrateur et les routeurs (Figure V.7 (a)), ainsi qu'entre les routeurs et ses nœuds attachés (Figure V.7 (b)). Cette comparaison va nous permettre d'avoir une métrique sur les retards de transmission pour les deux cas analysés.

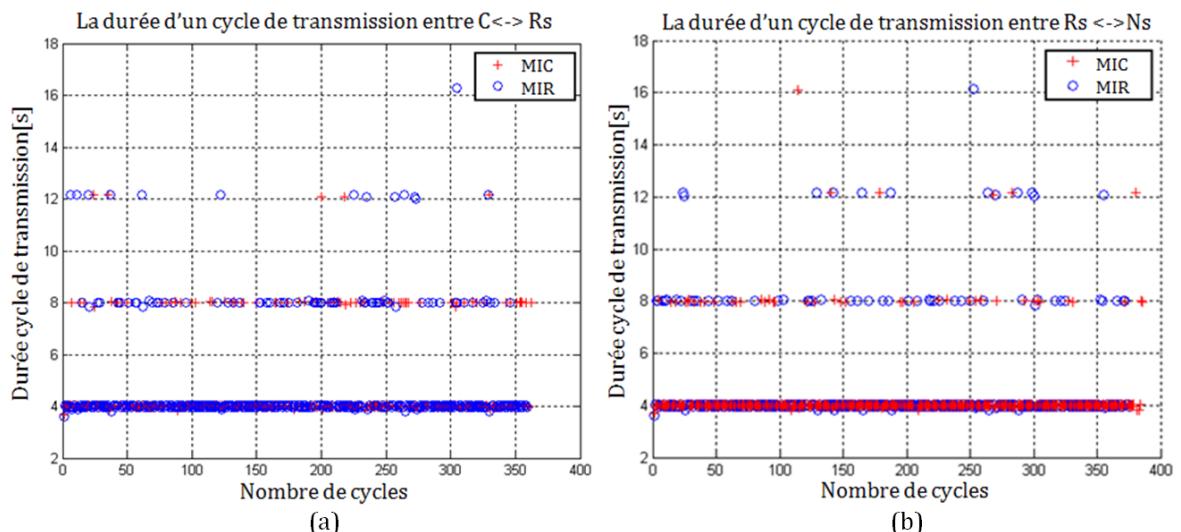


Figure V.7 Durée cycle de transmission entre C-Routeurs et R-Nœuds.

En analysant ces figures, nous pouvons constater que pour les deux niveaux de communication, l'organisation de type MIC présente un nombre plus petit des cycles avec une durée élevée, par rapport à l'organisation MIR (les mesures les plus grandes sont

principalement associées à la configuration MIR). Par conséquent, l'organisation MIC a globalement des délais de transmission plus faible que l'organisation de type MIR. Donc, même si les valeurs obtenues restent proches, nous pouvons affirmer que notre proposition (MIC) pour l'organisation du système paraît plus fiable, surtout si le nombre des éléments du réseau augmente (passage à l'échelle).

ÉVALUATION DE LA CONSOMMATION D'ÉNERGIE

Dans ce paragraphe, nous nous intéressons à la consommation d'énergie de nos systèmes, en comparant l'impact de la topologie. Pour s'approcher le plus possible du cas réel, nous avons utilisé comme valeurs initiales pour les paramètres de consommation d'énergie celles correspondantes à un nœud capteur Berkeley Mote [147]. Ainsi, la consommation d'énergie pour envoyer un message sera de 7mW et pour recevoir un message de 4.5mW. En ce qui concerne l'énergie totale de chaque nœud, elle sera initialisée à 2700J, ce qui correspond à une cellule CR2032 (3V, 230mA).

Les résultats obtenus en simulation (Figure V.8), montrent pour le concentrateur une consommation en énergie comparable pour les deux scénarios qui font l'objet de notre étude. Aussi, sachant que cet élément bénéficie d'une alimentation en continu, nous avons considéré peu significatif de prendre en compte cet aspect dans notre analyse.

Concernant les éléments actifs (les nœuds capteurs), nous avons trouvé, après une étape d'extrapolation des résultats, une différence entre la durée de vie de la batterie pour les deux scénarios, de 4,9% dans l'avantage de la solution MIC (Figure V.8 (a)).

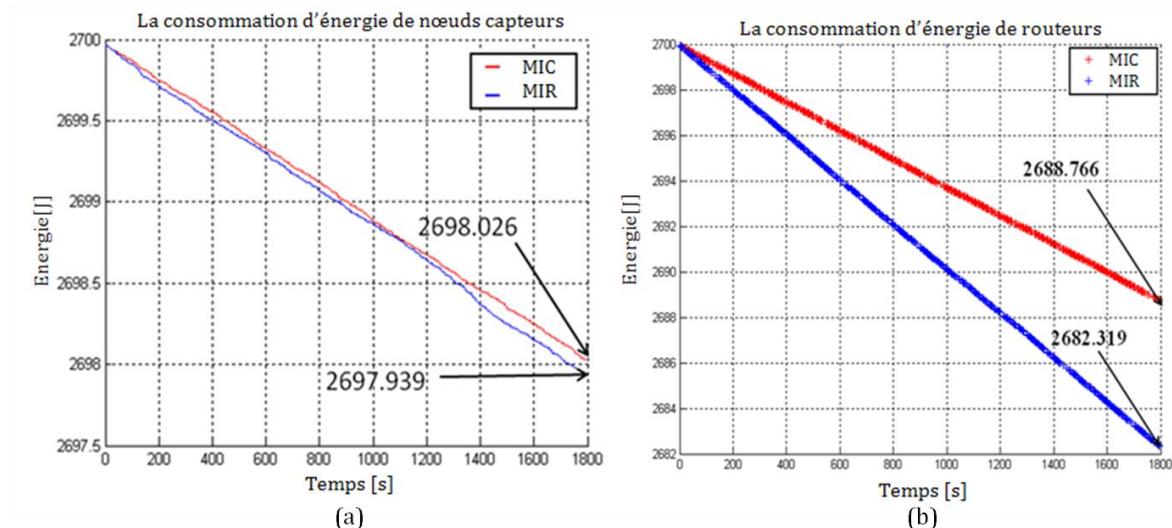


Figure V.8 La consommation d'énergie pour les nœuds capteurs pour les routeurs.

Le gain en énergie que nous avons obtenu ci-dessus pour l'organisation de type MIC, semble peu important. Mais, pour des éléments limités en énergie et qui doivent fonctionner le plus long temps possible, des moyens modestes en termes d'économie d'énergie comme celui-ci, peuvent contribuer significativement à la prolongation de la durée de vie du système.

Nous avons appliqué les mêmes méthodes (l'algorithme de simulation et une extrapolation des résultats) pour calculer la consommation d'énergie des routeurs. Ainsi, la durée de vie de la batterie d'un routeur du scénario de type MIC est avec 57% supérieure à celle d'une batterie identique de la configuration MIR (Figure V.8 (b)). Cette différence en consommation est due à la construction d'un routeur : dans une organisation de type MIR,

cet élément est équipé avec 2 interface de communication, ce qui va entraîner de point de vue routage, une augmentation importante du trafic. Au contraire, dans d'organisation de type MIC, le routeur détient une interface de communication, ce qui est équivalent avec un trafic de routage divisé par deux par rapport à l'autre solution.

Cette valeur montre que l'ensemble de l'architecture de communication d'un élément a une influence directe sur la consommation d'énergie du système entier. Concrètement, dans le scénario de type MIR (Figure V.5), la présence des deux interfaces de communication au niveau des éléments intermédiaires (les routeurs), va augmenter considérablement le trafic, ce qui se traduit par une dépense énergétique multipliée par deux ou plus.

Autrement dit, ce n'est pas que le choix de la pile de protocoles qui est important pour rendre le système plus fiable, mais aussi la conception du point de vue topologie. Ainsi, à travers tous les résultats présentés ci-dessus, nous pouvons justifier pourquoi une organisation de type MIC peut être plus adaptée pour un réseau de capteurs sans fil.

V.2.3 Synthèse

Tout au long de cette partie, nous avons évalué un réseau de communication avec une organisation hiérarchique, architecture basée sur un protocole MAC-TDMA et l'avantage de l'utilisation de multiples interfaces de communication. Pour pouvoir valider notre proposition, nous avons conçu pour cette architecture réseau un modèle de simulation sous NS-2.

À travers des résultats de simulation, nous avons montré que le nombre d'éléments intermédiaires (les routeurs) jouent un rôle important dans la réduction des retards de transmission. Également, pour réduire les coûts de fabrication d'un grand réseau, il est nécessaire que les éléments du système soient construits de la manière la plus homogène possible. Dans cette idée, nous avons proposé un modèle de réseau dans lequel seul le concentrateur est équipé d'interfaces multiples (l'organisation de type MIC).

Ainsi, notre solution présente un certain nombre d'avantages. Le premier est que la précision de la synchronisation n'est pas altérée. Le deuxième avantage est que, en termes de stabilité des délais de transmission, notre solution est plus efficace, ce qui représente un atout pour un éventuel passage à l'échelle. Le troisième et peut être le plus important, est que notre proposition d'architecture du système est capable de réaliser d'importantes économies d'énergie, ce qui a un impact positif sur la durée de vie du réseau.

Donc, à travers cette étape de conception, nous sommes en mesure de proposer une architecture réseau qui répond aux besoins et aux contraintes de notre application. Ainsi, nous pouvons traiter par la suite un des points durs de ces types d'applications, la précision de la synchronisation de données.

V.3 La synchronisation vs. l'économie d'énergie

Dans cette partie, nous nous sommes intéressés à la condition la plus importante de l'application, la synchronisation et la précision des données dans une très petite fenêtre

temporelle (de l'ordre de la nanosecondes), qui permettront de mieux corréler les données recueillies et d'améliorer la fluidité et la dynamique de notre modèle de communication.

Etant donné le contexte, les WSNs, il est obligatoire d'assurer les performances de synchronisation sans diminuer la durée de vie du réseau. Les solutions existantes jusqu'à présent dans la littérature qui prennent en compte ces deux aspects (la consommation d'énergie et la synchronisation), ne sont pas suffisantes en termes de performances pour notre application (Section I.3.2.2). Ainsi, notre stratégie a été d'exploiter les performances de protocoles de synchronisation pour les réseaux filaires en les adaptant du point de vue consommation énergétique pour les WSNs.

Nous avons donc commencé par une étape de tests et d'expérimentations dans le but d'évaluer l'impact d'une liaison sans fil sur le fonctionnement du protocole IEEE-1588 ou PTP (Precision Time Protocol), étude qui fait l'objet de la section suivante. Puis nous avons proposé un nouveau protocole de synchronisation IEEE1588-PBS hybride, que nous évaluons dans la section V.3.2

V.3.1 Evaluation des solutions existantes

Dans cette partie, nous étudions le comportement du protocole IEEE-1588 sur une plateforme de communication sans fil WiMedia. Ce protocole standardisé en 2002, est aujourd'hui le seul capable de garantir sur des liens filaires symétriques, de très bonnes performances pour la précision de la synchronisation (des dizaines de nanosecondes).

A ce sujet, nous avons trouvé dans la littérature plusieurs expérimentations réalisées avec le standard IEEE-1588 pour des configurations classiques de systèmes sans fil. C'est le cas de [148], où les auteurs proposent un algorithme de synchronisation amélioré calculant et intégrant l'asymétrie du lien de communication, algorithme qui est censé augmenter la précision de la synchronisation des horloges. Une autre étude du standard IEEE-1588 sur la dérive des fréquences esclaves a été accomplie dans [149].

La précision de la synchronisation dans l'algorithme du PTP est influencée principalement par un paramètre nommé OWD (One-Way-Delay). Réaliser une bonne estimation d'OWD n'est pas une tâche facile, du fait qu'il peut être affecté par l'incertitude de la synchronisation. Ainsi, De Vito L. et al. proposent un état de l'art dans [111] des différentes solutions proposées dans la littérature pour la mesure d'OWD.

En se basant sur ces études, nous avons réalisé notre plate-forme d'expérimentation avec une mise en œuvre du protocole IEEE-1588 en présence de différentes topologies réseau. Sur cette base, nous avons effectué des mesures de performance, à travers des paramètres caractéristiques tel que le CSP (Clock Synchronization Precision).

V.3.1.1 *Description de la plateforme*

Pour étudier ce problème de précision de la synchronisation, nous avons conçu une plate-forme de tests comme présenté dans la Figure V.9, qui consiste en: 4 machines, 4 cartes NI PCI-1588, 2 cartes Wisair USB sans fil PCI Express Mini (qui utilisent la technologie UWB) et un oscilloscope LeCroy WavePro Zi série 760 avec une largeur de bande de 6 GHz.

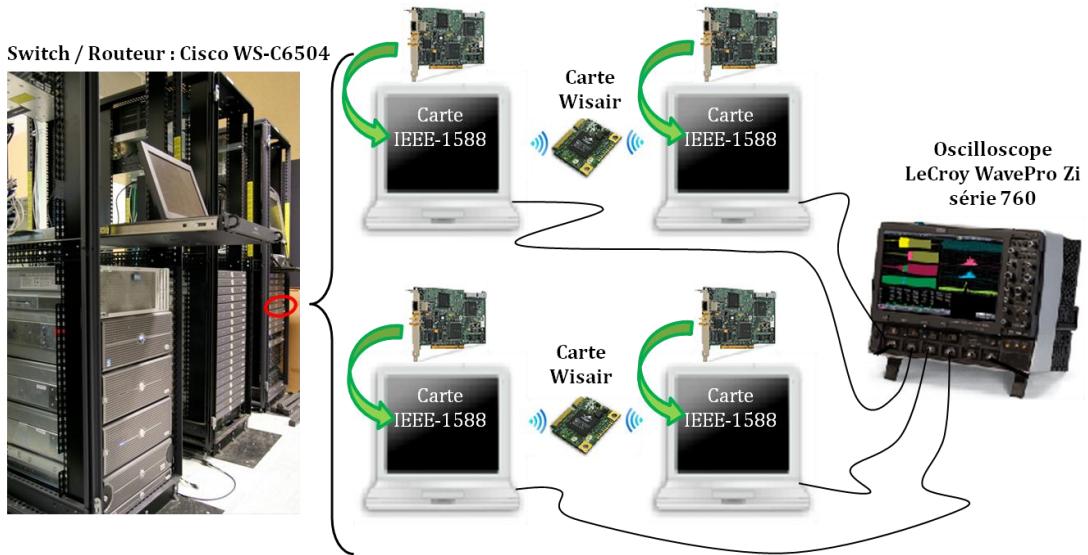


Figure V.9 Plate-forme d'expérimentation.

Les types de tests que nous avons effectués sont:

- ❖ **En connexion filaire** – mesure et validation des performances du protocole IEEE-1588, en concordance avec celles présentées dans le Data Sheet³⁹ de cartes;
- ❖ **En connexion sans fil** – mesures des performances du protocole IEEE-1588 en utilisant les cartes Wisair, basées sur la technologie de communication WiMedia ;
- ❖ **Application 1588** - pour une analyse plus détaillée de la connexion WiMedia, nous avons créé une application (un programme générateur de trafic, écrit en langage C) qui génère le même trafic qu'une interface IEEE-1588, comme décrit dans la Figure V.10 :

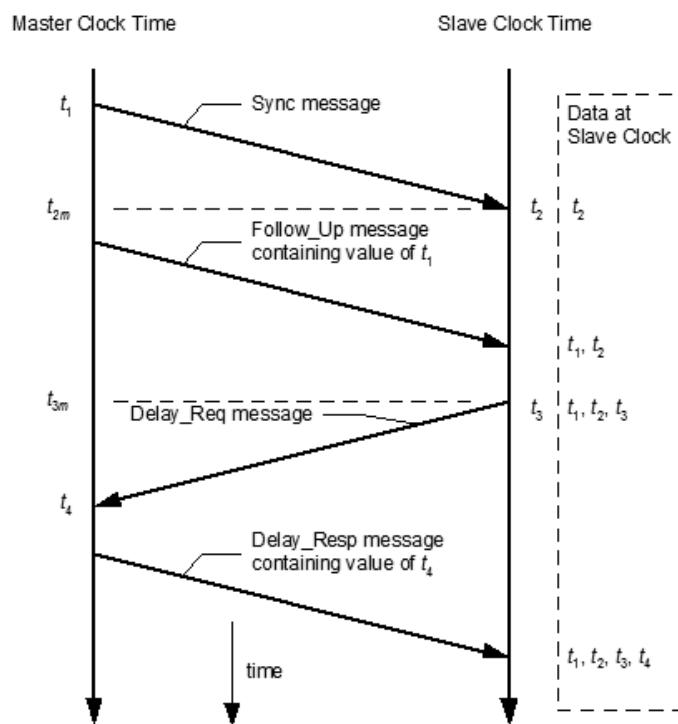


Figure V.10 Fonctionnement d'IEEE-1588⁴⁰.

³⁹ http://www.ni.com/pdf/products/us/cat_pci1588.pdf

En résumant cette figure, l'esclave synchronise son horloge avec celle de son maître à travers le processus de synchronisation qui est divisé en deux phases :

- ❖ La première corrige la différence de temps entre le maître et l'esclave, ce qui représente le décalage de temps ;
- ❖ La deuxième phase de la synchronisation mesure le retard, ça veut dire le temps de propagation entre l'esclave et le maître.

L'utilisation de cet algorithme de synchronisation permet d'éliminer les décalages entre les horloges internes (en particulier celles introduits par la pile protocolaire et le temps de latence entre le maître et l'esclave).

V.3.1.2 *Etudes sur les performances de synchronisation*

Nous allons maintenant analyser un certain nombre de résultats pour des tests réalisés dans des configurations filaires et sans fil. Ainsi, nous avons décidé d'effectuer trois campagnes différentes de mesures de performances pour l'IEEE-1588 :

- ❖ Analyse des lectures d'oscilloscope - pour obtenir la variation de la différence de phase entre les signaux carrés générés par les cartes IEEE-1588, c'est à dire le décalage des horloges synchronisés ;
- ❖ Analyse Matlab⁴¹ – pour calculer à partir des signaux captés par l'oscilloscope la précision de la synchronisation. Nous avons adopté cette deuxième méthode pour pouvoir vérifier si les résultats obtenus dans ces deux campagnes de tests étaient similaires avec ceux obtenus après l'analyse d'oscilloscope ;
- ❖ Génération de trafic et capture (Application 1588) – pour réaliser une description détaillée des paramètres du réseau (la gigue, le débit, le type de connexion), ce qui nous permettra de mieux comprendre l'influence du lien de transmission sur la qualité de services.

ETUDE 1 : CONNEXION FILAIRE

Les mesures réalisées sur la plate-forme filaire nous ont fourni des valeurs précises sur la capacité de synchronisation réelle de cartes IEEE-1588. Pour cela, nous avons mesuré la variation de la gigue entre deux signaux carrés envoyés par une sortie PFI de chaque carte NI PCI-1588, comme présenté dans la figure suivante :

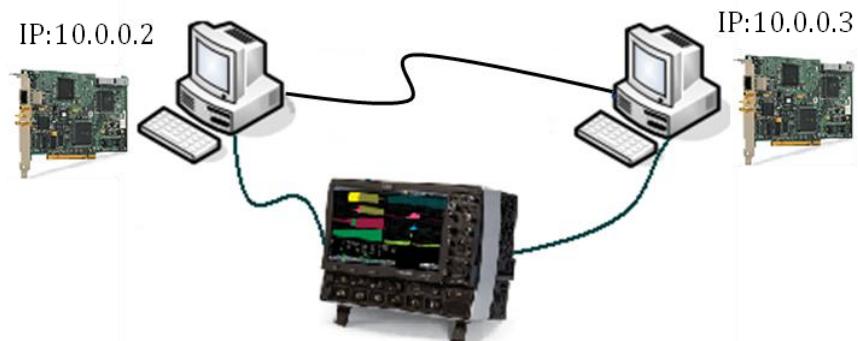


Figure V.11 Capture des signaux PFI sur l'oscilloscope (en filaire).

⁴⁰ http://www.real-time-systems.com/ieee_1588/index.php

⁴¹ <http://www.mathworks.com/products/matlab/>

Dans notre évaluation des performances, nous avons introduit plusieurs indicateurs: la valeur moyenne, l'écart type et le coefficient de variation (CV) [150], ainsi que la variation de la gigue (Λ). Le CV est utilisé pour évaluer la dispersion relative autour de la moyenne.

La première méthode pour évaluer la variation de la gigue, le Λ_1 , implique l'utilisation de la valeur moyenne d'une capture d'oscilloscope. Ainsi, pour trouver la valeur moyenne de chaque lecture, nous avons soustrait de cette valeur la valeur instantanée de cette variation, le Δ_i , pour les deux signaux (maître et esclave) enregistrés par l'oscilloscope (équation V.3).

$$\Lambda_1 = \text{moyenne}(|\Delta_{moy} - \Delta_i|) \quad (\text{V.3})$$

La deuxième méthode pour évaluer la variation de la gigue, le Λ_2 , consiste à calculer la différence entre deux valeurs consécutives du Δ (c'est à dire la gigue entre les deux signaux PFI) de chaque capture, suivit par le calcul de cette moyenne (équation V.4).

$$\Lambda_2 = \text{moyenne}(|\Delta_{i+1} - \Delta_i|) \quad (\text{V.4})$$

où Δ_i et Δ_{i+1} représentent les variations instantanées pour deux instants consécutifs.

Dans la Figure V.12, nous pouvons visualiser le décalage existant entre deux signaux générés par des cartes IEEE-1588. Cette différence de phase (Δ) ne montre pas de désynchronisation des horloges, mais seulement un décalage dans le démarrage du processus de synchronisation sur chaque poste. Ainsi, le paramètre CSP est ici la variation de cette différence de phase, que nous l'appellerons Λ_1 (méthode 1) ou Λ_2 (méthode 2).

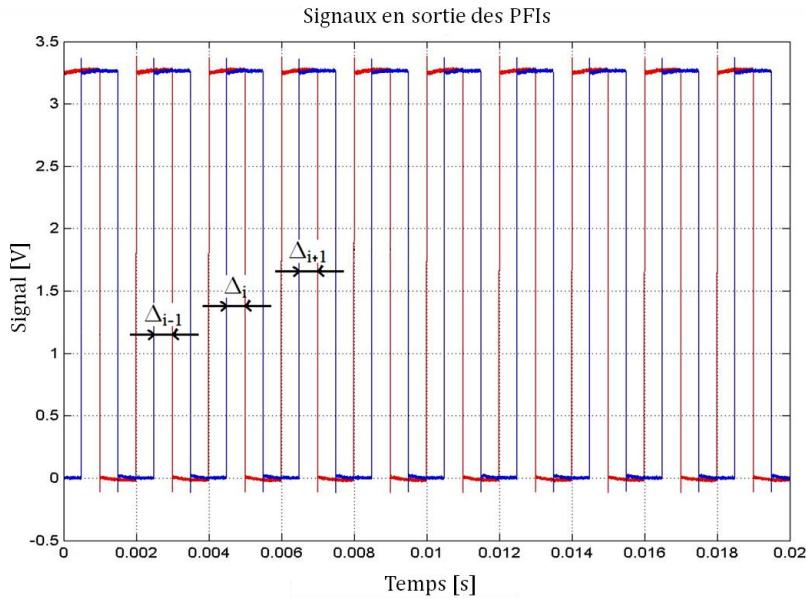


Figure V.12 Signaux en sortie des PFI.

Nous notons que nous avons réalisé plusieurs captures des signaux avec une durée de plusieurs dizaines de minutes, captures que nous avons par la suite analysées. Ainsi, concernant le premier scénario (Figure V.11), l'erreur moyenne de la variation du délai de la gigue (Λ_1, Λ_2) est de l'ordre de quelques dizaines de nanosecondes (Figure V.13) pour les deux méthodes d'évaluation, ce qui reste conforme aux valeurs fournies par le Data Sheet des cartes NI PCI-1588.

Cependant, nous pouvons constater que, en utilisant la méthode numéro 2, nous trouvons une grande variation du paramètre Λ_2 .

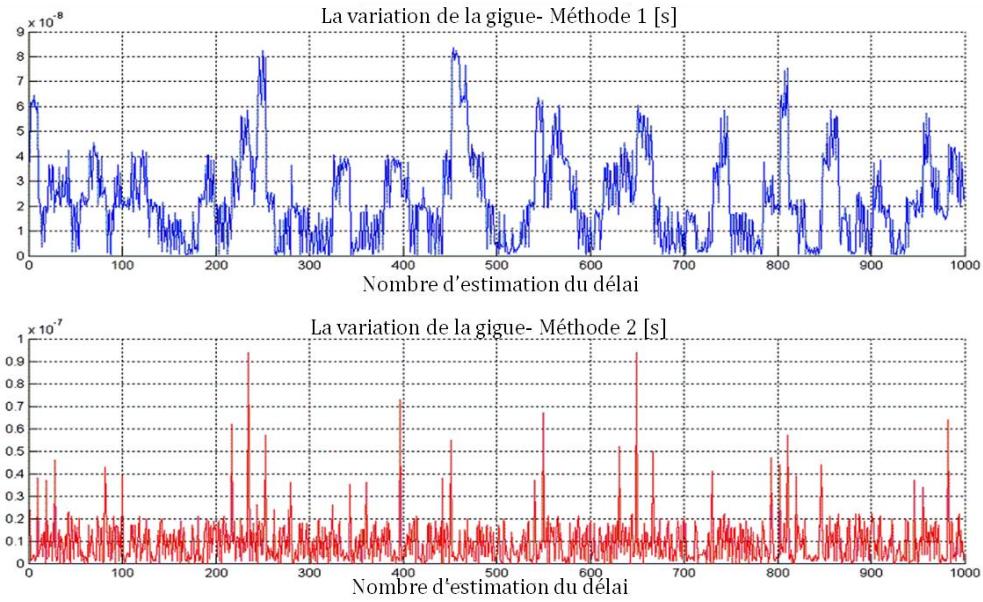


Figure V.13 Mesures de la variation de la gigue en connexion filaire (une seule capture).

Ce phénomène s'explique par l'évaluation de la valeur du coefficient de variation (CV), qui est d'environ 110% pour la méthode 2, contrairement à la méthode 1, où le CV est d'environ 80% (Figure V.14).

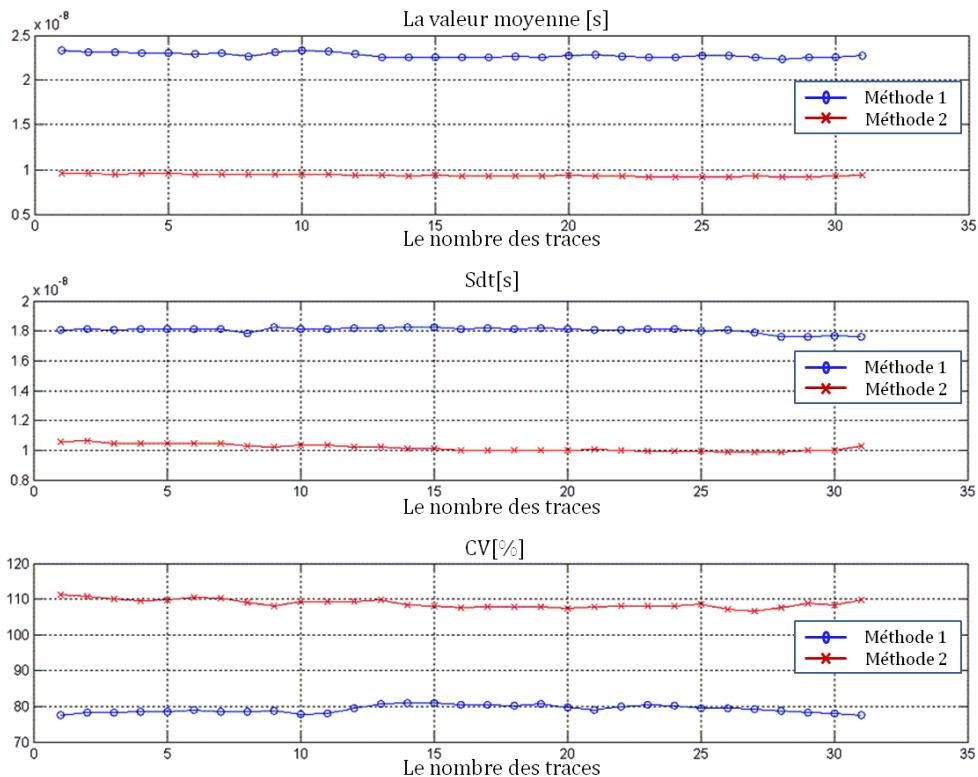


Figure V.14 Analyse de performances pour la variation de la gigue (~55 captures).

Nous pouvons donc constater que la valeur moyenne du paramètre Λ est plus proche de l'estimation du Data Sheet pour la première méthode (environ 24 ns), car pour la deuxième méthode nous avons trouvé une moyenne d'environ 10 ns. D'après ces résultats,

nous supposons que la méthode de mesure adoptée dans le catalogue correspond à notre première méthode d'évaluation pour la variation de la gigue. Par conséquent, nous continuerons nos expérimentations en utilisant que la méthode 1.

ETUDE 2 : CONNEXION SANS FIL (WiMEDIA)

Dans le but d'analyser les performances du protocole IEEE-1588 dans une configuration sans fil, nous avons réalisé les mêmes mesures, mais cette fois-ci à travers une transmission assurée par des cartes Wisair, et représentée dans la Figure V.15.



Figure V.15 Capture des signaux PFI sur l'oscilloscope (en sans fil).

Ainsi, après l'analyse de captures réalisées avec l'oscilloscope, nous avons trouvé une valeur moyenne du délai de la gigue de l'ordre de quelques centaines de microsecondes (Figure V.16 et Figure V.17).

Cette différence entre les résultats obtenus sur les deux types de connexions (de 24 ns sur filaire contre une centaine de μ s sur sans fil), nous a convaincu de faire une analyse plus détaillée du comportement d'IEEE-1588 en présence de cartes WiMedia.

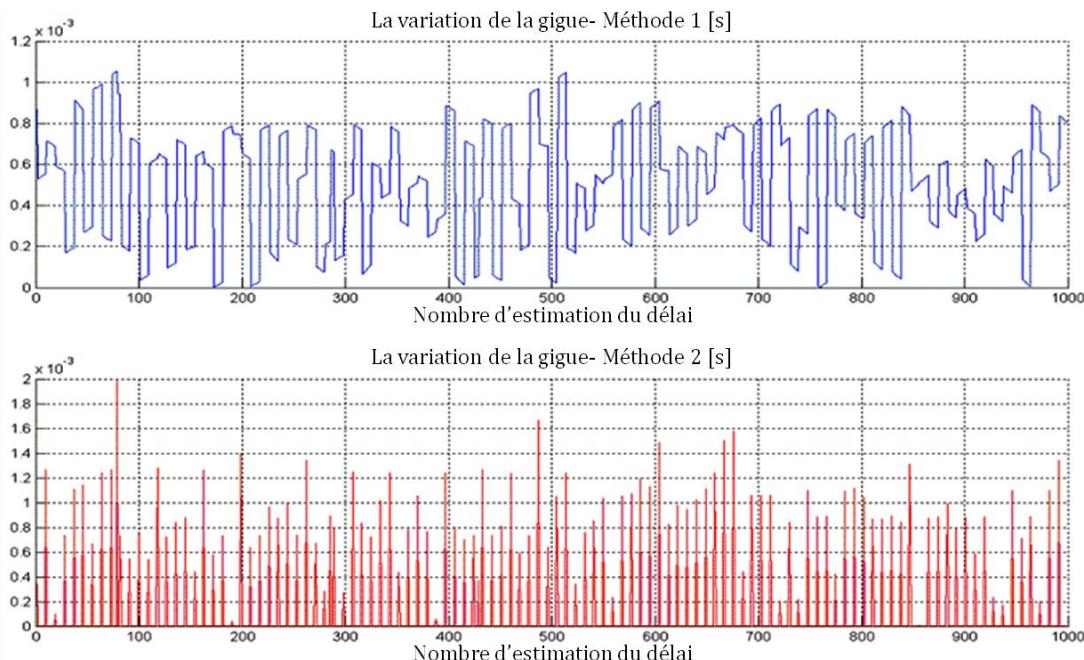


Figure V.16 Mesures de la variation de la gigue en sans fil (deux méthodes, une capture).

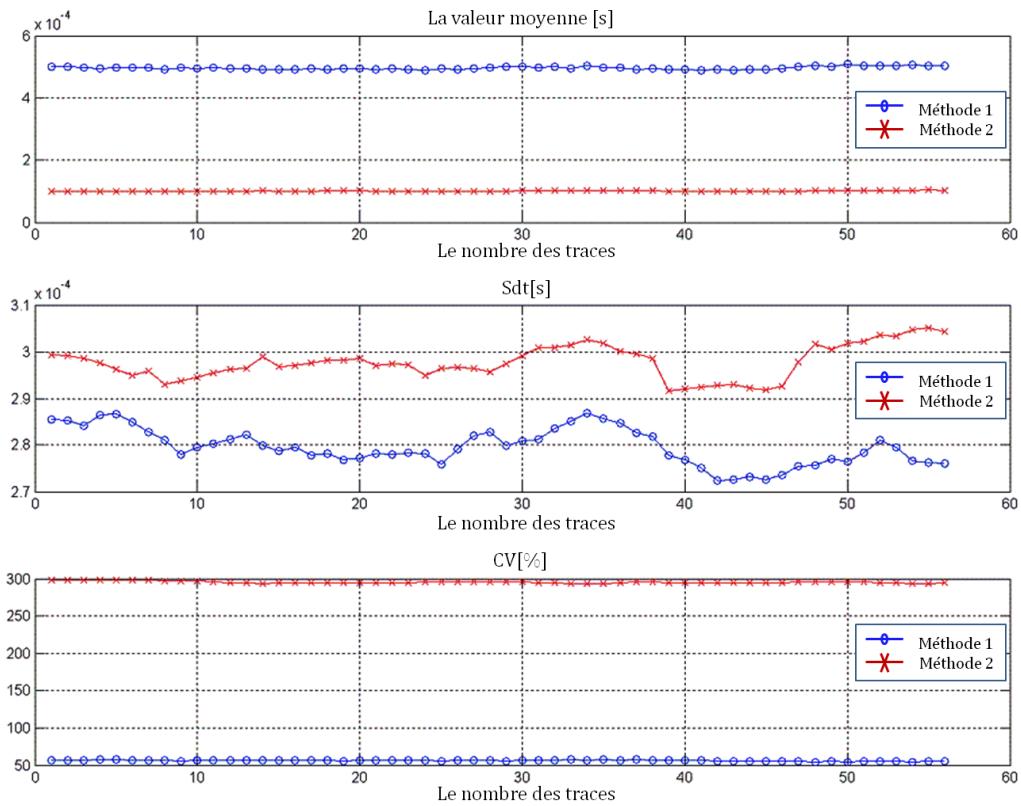


Figure V.17 Analyse de performances pour la variation de la gigue (~55 captures).

Comme pour le cas filaire, nous constatons que la première méthode d'évaluation (Figure V.16) présente moins de variations pour la variation de la gigue par rapport à la méthode 2, ce qui se traduit par une valeur moyenne du Δ plus élevée et une valeur du CV moins élevée pour la méthode 1, par rapport à la méthode 2.

Ainsi, nous concluons que la précision de la synchronisation (CSP) est moins performante que dans une connexion filaire et peut-être insuffisante pour les réseaux de capteurs sans fil.

TRAITEMENT MATLAB (FILAIRE ET SANS FIL)

Pour vérifier l'exactitude de résultats obtenus ci-dessus, nous avons récupéré sous forme de fichiers Matlab les signaux d'origine capturés par l'oscilloscope pour pouvoir calculer la variation du retard entre le front montant des signaux C1 et C2. L'avantage de ce type d'analyse est qu'il permet d'enlever les possibles erreurs introduites par l'oscilloscope. Nous notons que nous avons utilisé la méthode 1 comme méthode d'évaluation.

Dans cette campagne de validation de résultats, nous obtenons pour le paramètre Δ une valeur de l'ordre de 0.5×10^{-3} ns dans une connexion filaire (Figure V.18), et de $4,5 \mu\text{s}$ pour une connexion sans fil (Figure V.19).

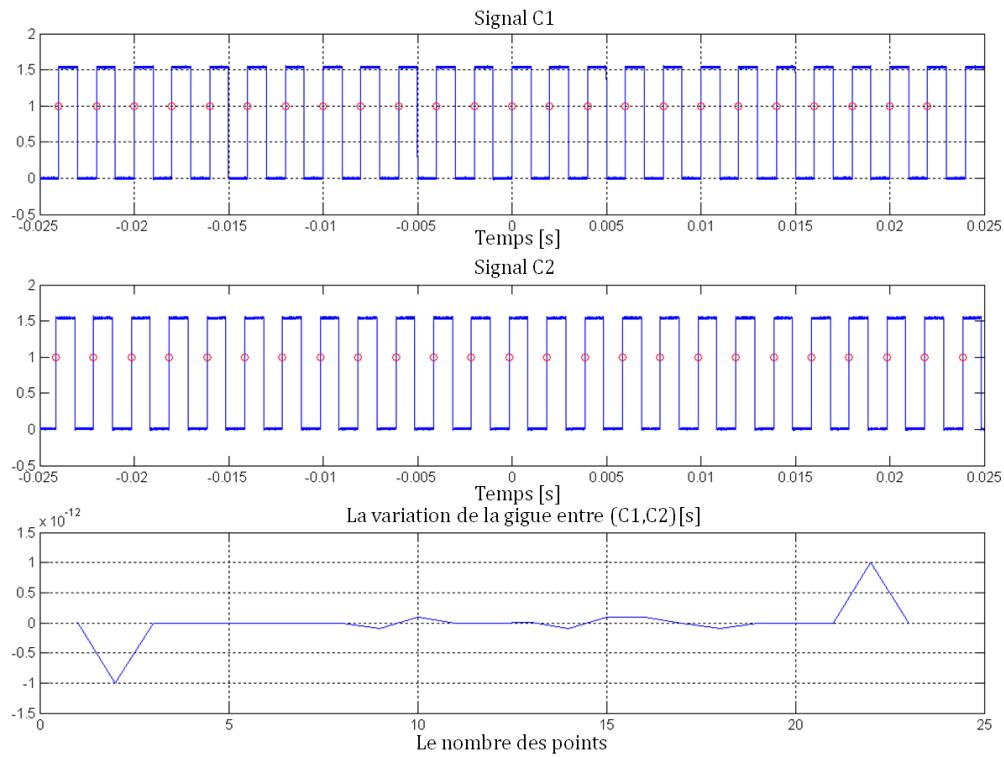


Figure V.18 Signaux et la variation de la gigue pour une connexion filaire (méthode 1).

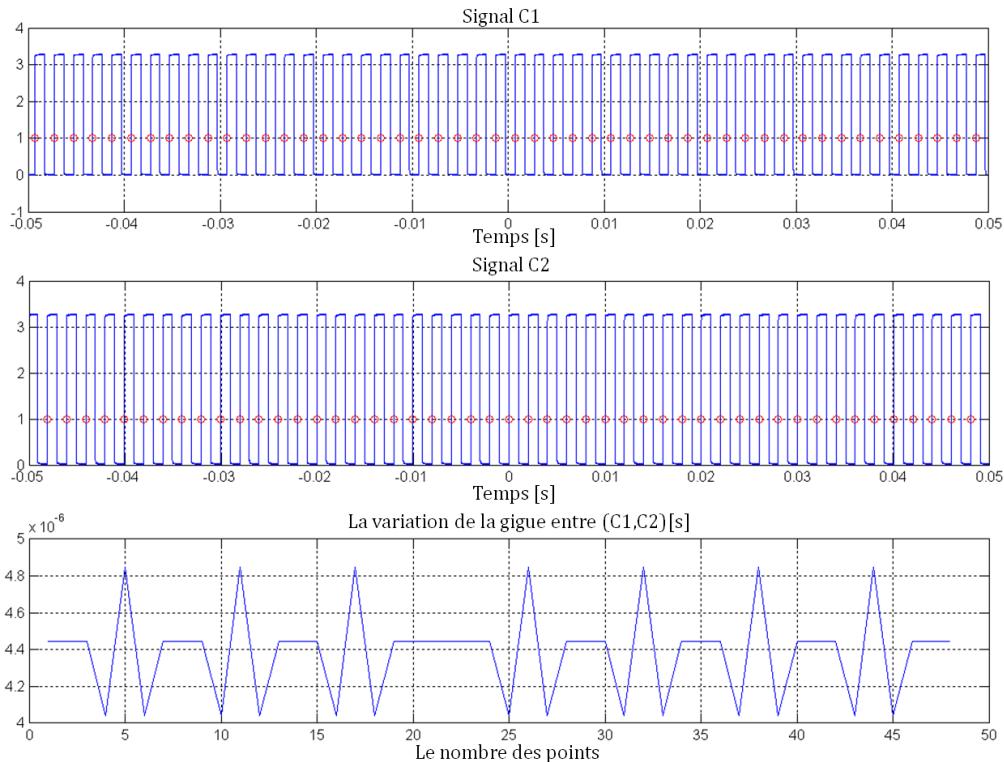


Figure V.19 Signaux et la variation de la gigue pour une connexion sans fil (méthode 1).

La différence entre les valeurs obtenues pour la variation de la gigue dans ces deux campagnes de mesures (Tableau V.1) s'explique par les erreurs de calcul introduites par

l'oscilloscope, car pour le cas filaire la précision de la synchronisation mesurée est plus proche de la valeur de référence (33-75ns).

| Type de mesures | Précision de la synchronisation | |
|---|---------------------------------|-------------------------|
| | Analyse par oscilloscope | Analyse Matlab |
| Campagne de mesures en connexion filaire | 24 ns | 0.5×10^{-3} ns |
| Campagne de mesures en connexion sans fil | Une centaine de μ s | 4,5 μ s |

Tableau V.1 Comparaison de valeurs obtenues pour la précision de la synchronisation.

Nous pouvons aussi mentionner que l'analyse réalisée en Matlab confirme la dégradation des performances (CSP) du protocole IEEE-1588 dans une connexion sans fil. Cependant, nous avons décidé de caractériser et d'analyser l'influence du lien de communication sur la transmission des données dans une configuration filaire, ainsi que sans fil, pour pouvoir expliquer cette dégradation de performances pour le sans fil.

Par la suite, nous allons présenter l'application 1588 que nous avons créée pour analyser l'échange des messages et pour mieux caractériser le lien de communication, en connexion filaire et sans fil.

APPLICATION 1588 (FILAIRE ET SANS FIL)

Le générateur de trafic IEEE-1588 (Figure V.20) a été créé dans le but d'étudier les problèmes de retard moyen du réseau et pour minimiser les éventuelles erreurs de mesures et de calculs, mais aussi pour mieux comprendre les délais de transmissions en estampillant les messages de type IEEE-1588.

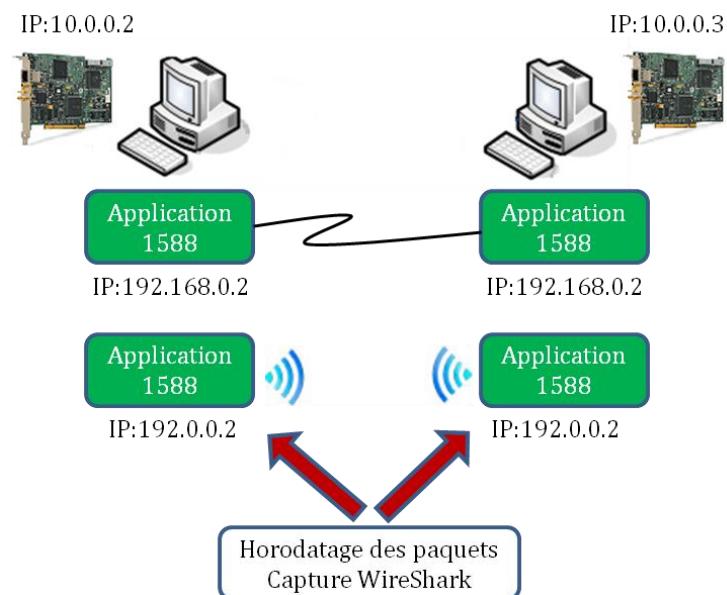


Figure V.20 Plate-forme de test filaire et sans fil.

Ainsi, l'application génère le même trafic que le protocole IEEE-1588 (la même suite de paquets Sync, Follow_up, Delay_req, et Delay-resp, en respectant les intervalles de temps (Figure V.10)). Dans ces conditions, nous avons pris des captures avec le logiciel d'analyse de trafic Wireshark⁴².

⁴² <http://www.wireshark.org/>

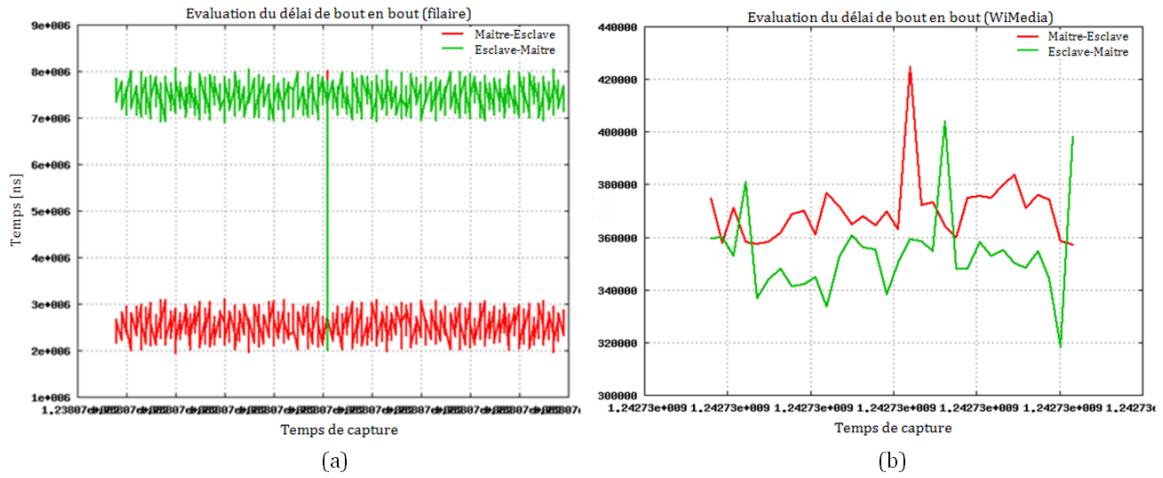


Figure V.21 Délai entre Maitre-Esclave et Esclave-Maitre sur une liaison filaire et sans fil.

Nous observons en analysant ces deux dernières figures (Figure V.21 (a) et (b)), que la variation du délai (la gigue) a une valeur comparable entre les deux types de connexions, filaire et sans fil (0,5 μ s, respectivement 0,35 ms). Concernant la différence du délai entre le flux montant et le flux descendant (entre maître et l'esclave), elle a une valeur de 5 μ s en connexion filaire et de 40 ms en sans fil (WiMedia). Nous rappelons que le protocole IEEE-1588 fait l'hypothèse d'un lien symétrique et qui ne calcule pas le paramètre OWD.

De plus, pour une capture de plusieurs dizaines de minutes, les mesures montrent l'influence de l'asymétrie du medium de communication sur la précision de la synchronisation offerte par le protocole IEEE-1588. Nous avons pu mesurer une asymétrie de 6 ms de délai entre la voie aller et la voie retour sur le kit WiMedia de Wisair, alors qu'elle est d'environ 100 μ s sur le lien Ethernet. Cette asymétrie s'explique par le protocole MAC de la carte Wisair qui utilise le paradigme maître esclave avec une forte priorité pour le maître, pour l'accès au réseau.

V.3.1.3 Conclusion

Nous avons vu que la condition essentielle pour ce projet serait d'avoir une synchronisation inférieure à la microseconde pour les données recueillies par les nœuds capteurs.

Suite à cette campagne de tests, nous avons obtenu une série de résultats qui seront utilisés dans notre application de réseaux de capteurs sans fil. Ainsi, une première conclusion est que nous avons montré le caractère asymétrique du lien de communication, ce qui explique la dégradation de performances du protocole IEEE-1588 pour une connexion sans fil.

Un autre paramètre que nous avons analysé dans le cadre de la comparaison Ethernet/WiMedia, la gigue, montre des valeurs comparables pour les deux méthodes de connexion, de l'ordre de centaines de microsecondes, ce qui ne représente pas un facteur majeur dans la dégradation des performances de synchronisation en présence de cartes Wisair.

Nous notons qu'en théorie il est possible d'obtenir de bonnes performances pour le protocole IEEE-1588 sur WiMedia (qui est conçu pour pouvoir faire de communications

l'isochrones). Par contre, en pratique, sur les cartes Wisair, ce concept n'est pas implémenté, ce qui se reflète dans la dégradation des performances.

Donc, la solution sera d'utiliser une couche MAC capable de garantir ces contraintes sur la gigue (max 100 μ s) et de symétrie de lien, afin de garantir le bon fonctionnement du protocole de synchronisation. Ainsi, ces valeurs seront prises en compte dans la conception des couches MAC et réseau de notre application.

Cependant, nous présentons par la suite les performances obtenues en simulation pour notre protocole de synchronisation, solution qui a été proposée suite aux résultats insatisfaisants de ces campagnes de tests et de mesures.

V.3.2 Notre proposition de synchronisation. Analyse comparative des performances

Etant donnés les résultats obtenus dans les campagnes de tests présentées auparavant, et compte tenu de l'analyse des protocoles de synchronisation proposés aujourd'hui dans la littérature, analyse réalisée dans la section I.3.2.2, nous avons conclu qu'aucune des solutions existantes n'est adaptée pour les contraintes de notre application. Donc, nous avons décidé de concevoir un nouveau protocole de synchronisation.

Ainsi, notre proposition est basée sur l'extension d'IEEE-1588 et du protocole PBS, algorithme que nous avons présenté dans le chapitre III. Nous rappelons ci-dessous (Figure V.22) le fonctionnement de notre protocole, l'IEEE1588-PBS hybride, pour un groupe réduit à seulement 3 nœuds :

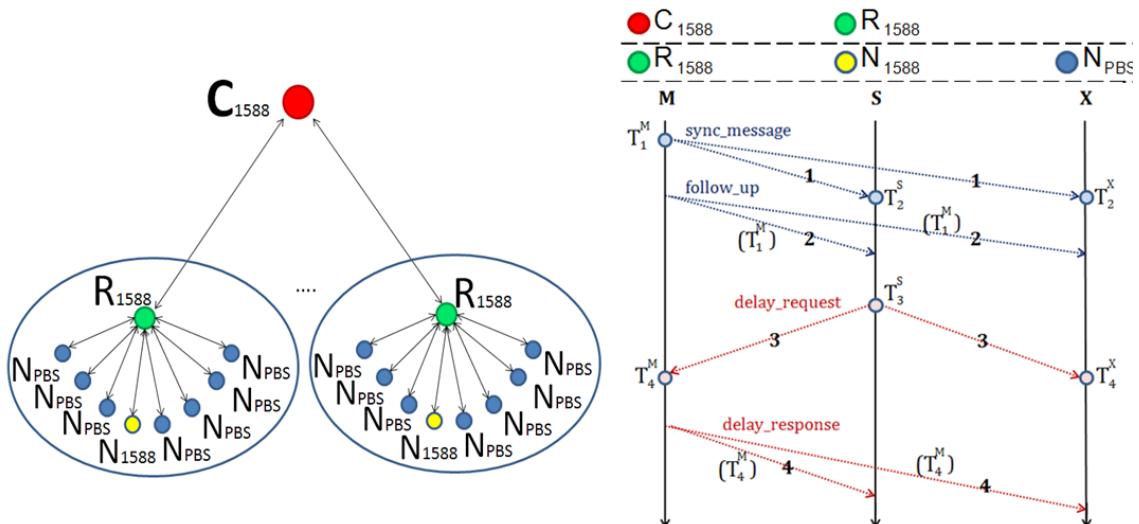


Figure V.22 Topologie réseau et fonctionnement du l'IEEE1588-PBS hybride.

Dans un tel groupe, le nœud maître (M) et le nœud esclave (S) se synchronisent en utilisant l'algorithme d'IEEE-1588, temps que le troisième nœud (X) va utiliser la synchronisation déjà établie entre le maître et l'esclave en écoutant les messages échangés (comme dans l'hypothèse du protocole PBS). Celle-ci va permettre au nœud X de synchroniser son horloge interne avec celle du nœud M.

Nous présentons ainsi dans la section suivante une analyse comparative entre les performances de notre protocole IEEE1588-PBS hybride et du standard IEEE-1588 obtenue

après l'implémentation des protocoles dans le simulateur de réseau NS-2. Les critères pris en compte dans cette évaluation sont les suivants:

- ❖ La précision de la synchronisation ;
- ❖ La consommation d'énergie;
- ❖ Le nombre de messages de synchronisation.

Pour vérifier et valider le comportement et les performances de la nouvelle proposition, il faut tout d'abord spécifier les principaux paramétrages considérés dans notre évaluation.

Ainsi, nous avons lancé le mécanisme de synchronisation pour le deuxième niveau (Routeurs-Nœuds) quelques secondes après le premier niveau (Concentrateur-Routeurs), ce qui va permettre aux éléments du Niveau 1 d'être déjà synchrones. Par conséquent, les nœuds pourront corriger leur horloge interne en prenant comme référence l'horloge des routeurs qui sont déjà synchronisés.

Nous avons déjà mentionné que dans un WSN, pour obtenir une bonne précision de la synchronisation, il est nécessaire que l'horloge interne soit contrôlée par un oscillateur à cristal de compensation en température (TCXO à 37.5 Mhz [53]), qui peut avoir une tolérance en fréquence de 1,5 ppm, permettant de réduire le taux de la dérive d'horloge. Ainsi, dans notre simulation, l'horloge interne de chaque nœud du réseau a été implémentée avec une dérive de 1,5 microseconde par seconde.

Pour obtenir des meilleures performances, et pour pouvoir comparer notre protocole avec la norme IEEE-1588, nous avons décidé d'horodater les messages de synchronisation au niveau de la couche physique. Ces informations seront ensuite utilisées dans l'algorithme de synchronisation au niveau de la couche applicative.

V.3.2.1 La précision de la synchronisation

Nous avons effectué des simulations pour l'ensemble du réseau (un concentrateur, 8 routeurs et 64 nœuds) (Figure V.23). Du fait que l'organisation de notre système soit hiérarchique, le comportement d'un sous-groupe (un concentrateur, un routeur et huit nœuds) est similaire à celui de tous les autres sous-groupes. Ainsi, nous allons présenter par la suite des résultats de simulations qui concernent un tel sous-groupe d'éléments.

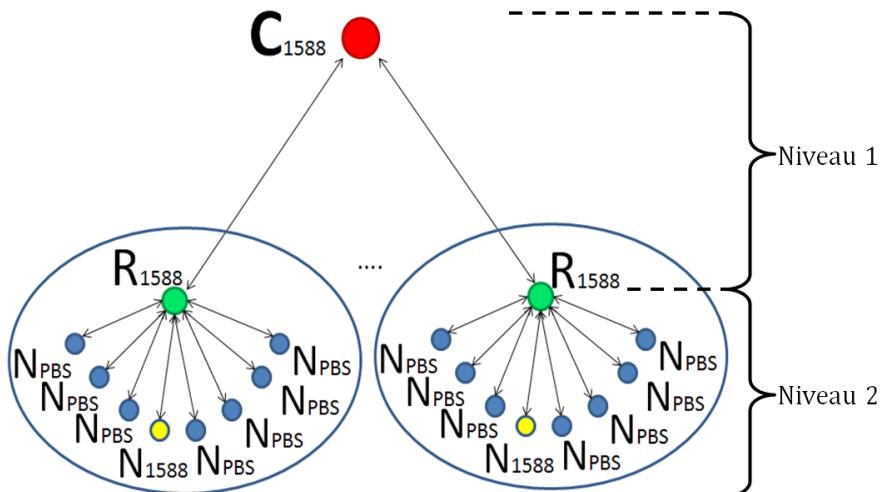


Figure V.23 Le réseau simulé.

Par conséquent, dans la Figure V.24 nous avons représenté la précision de la synchronisation pour le Niveau 1 de communication (Concentrateur - Routeurs), qui est de l'ordre de la centaine de nanosecondes. Nous rajoutons que, pendant toute la durée de simulation, les éléments du système restent synchronisés à un niveau élevé, entre 50 et 290ns.

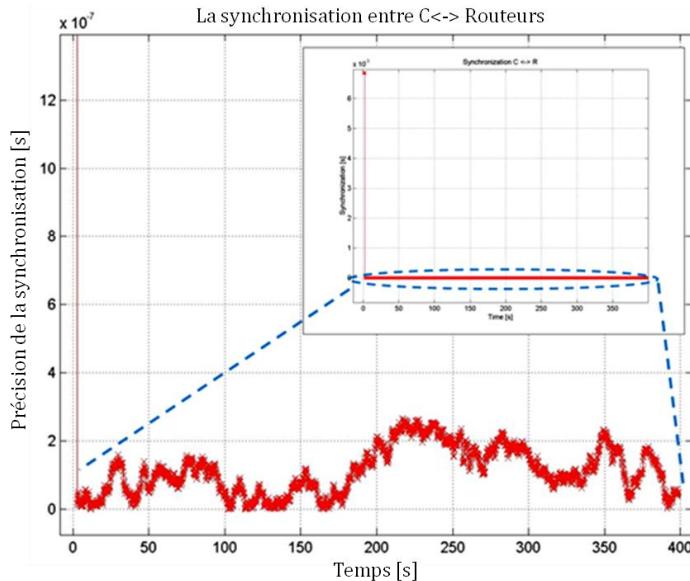


Figure V.24 La synchronisation entre C <-> Rs pour l'IEEE1588-PBS hybride.

De plus, il faut mentionner que les performances trouvées dans la littérature pour le protocole IEEE-1588 sur une liaison sans fil (~ 200 ns) sont en accord avec nos résultats, ce qui s'explique par le fait que nous avons respecté l'implémentation de l'algorithme de synchronisation, et les mêmes conditions de fonctionnement.

En ce qui concerne le Niveau 2 de communication (Routeurs - Nœuds), nous constatons que pour la paire Routeur-Node1588, la précision de la synchronisation a une valeur moyenne de 180 ns et que pour les paires Routeur-NodePBSs la valeur varie entre 5 μ s et 600 ns (Figure V.25).

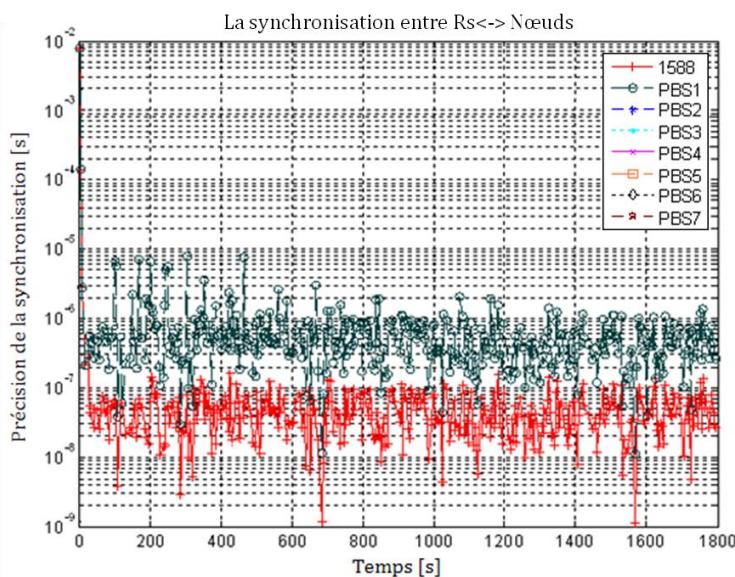


Figure V.25 La synchronisation entre Rs <-> Ns pour l'IEEE1588-PBS hybride.

Nous constatons que les résultats obtenus pour la paire Routeur-Node1588 sont comparables avec les performances du standard IEEE-1588 (~ 200 ns). Nous tenons aussi à préciser que la synchronisation globale du réseau est accomplie après environ 20 secondes du début de la simulation, ce qui s'explique par le fait que la synchronisation du deuxième niveau de communication et dépendante de l'état de synchronisation du premier niveau.

Par conséquent, nous montrons par ces résultats que la précision de la synchronisation obtenue pour notre protocole est très proche de meilleures performances trouvées dans la littérature.

V.3.2.2 *Evaluation de la consommation d'énergie*

Par la suite, nous nous intéressons à l'analyse de la consommation d'énergie dans notre système, et plus précisément à la consommation des nœuds capteurs. Pour s'approcher au plus près de conditions réelles, nous avons utilisé les mêmes paramètres de consommation énergétique que avant (l'énergie initiale d'un nœud de 2700J, l'énergie consommée pour envoyer un message de 7mW et pour recevoir un message de 4.5mW).

Nous avons analysé la différence entre la consommation d'énergie pour les deux protocoles étudiés (l'IEEE-1588 et l'IEEE1588-PBS hybride). Ainsi, la Figure V.26 montre que la durée de vie de la batterie d'un nœud PBS est prolongée avec 75% par rapport à la batterie d'un nœud 1588. Cette différence est directement liée au nombre de messages envoyés et reçus par chaque nœuds pour réaliser la synchronisation.

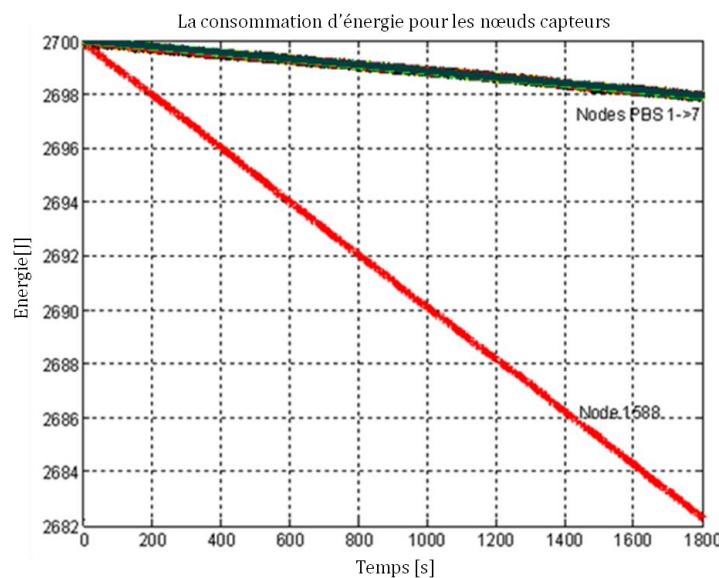


Figure V.26 La consommation d'énergie pour les nœuds du réseau.

Par conséquent, nous avons calculé (Tableau V.2) le nombre de messages de synchronisation utilisés par les deux protocoles analysés:

| Protocoles | Nombre de messages | |
|----------------------|--------------------|--------------------|
| | Niveau C<->Rs | Niveau Rs<->Ns |
| IEEE1588-PBS hybride | 120messages/cycle | 24messages/cycle |
| IEEE1588 | 120messages/cycle | 1024messages/cycle |

Tableau V.2 Les messages de synchronisation.

Cette analyse nous permet de comprendre pourquoi notre solution est plus appropriée pour un réseau de capteurs sans fil. Ainsi, nous avons montré que notre solution assure une bonne précision pour la synchronisation et atteint en même temps des économies non négligeables au niveau énergétique.

V.3.3 Conclusion

Le protocole de synchronisation que nous avons proposé et simulé en NS-2 repose sur les principes de fonctionnement du standard IEEE-1588 et du protocole PBS. Les résultats ainsi obtenus montrent que cette solution est capable d'offrir une précision de synchronisation de l'ordre de centaines de nanosecondes et en même temps de faire des économies d'énergie, contrairement à d'autres protocoles étudiés auparavant.

Dans la suite de ce travail, nous nous proposons de regarder de plus près l'optimisation énergétique dans le cas particulier d'un service de synchronisation d'horloge, en l'adaptant à l'utilisation d'antennes intelligentes, qui ont la capacité d'améliorer les gains d'antenne et donc les besoins en puissance.

V.4 La directivité des antennes vs. l'économie d'énergie

Dans le chapitre IV, nous avons réalisé une présentation succincte des technologies d'antennes intelligentes, nous avons introduit la problématique soulevée par ces antennes et nous avons proposé une solution. L'objectif de cette partie est, par conséquent, d'évaluer cette proposition en simulation, à l'aide du simulateur NS-2.

V.4.1 Analyse comparative des performances

Le principal avantage des antennes directives est leur propriété de focaliser le spectre de rayonnement en direction du récepteur, principe valable aussi pour l'antenne de réception. Ces antennes peuvent être fixes et nous parlons d'antennes directives, ou orientables à la demande et nous parlons alors d'antennes intelligentes. Comme notre exemple d'application aura des capteurs statiques, nous avons considéré dans notre étude le premier type d'antennes.

Contrairement aux émissions par antennes omnidirectionnelles, le signal est concentré sur le récepteur et de fait les interférences entre émetteurs sont réduites. L'augmentation du SNR induit une augmentation de la bande passante disponible ou un accroissement de la portée du réseau. Ainsi, pour un SNR constant, un émetteur peut baisser sa puissance d'émission sans dégrader la qualité par rapport à une émission omnidirectionnelle et ainsi réduire sa consommation énergétique.

Nous présentons donc par la suite des évaluations de performances et une étude comparative entre notre solution Directionnal-TDMA MAC et le MAC-TDMA initial en présence d'antennes omnidirectionnelles.

L'IMPACT SUR LA PRECISION DE LA SYNCHRONISATION

Le besoin de synchronisation est primordial pour notre application et il est donc impératif d'évaluer ce paramètre en présence de la nouvelle pile de protocoles.

Comme nous l'avons vu précédemment, notre réseau présente deux niveaux hiérarchiques de communication (Concentrateur-Routeurs et Routeurs-Nœuds). Pour cette raison, nous débutons le mécanisme de synchronisation toujours avant pour les paires Concentrateur-Routeur par rapport aux paires Routeur-Nœuds. Ainsi, les nœuds bénéficieront de l'horloge du routeur comme référence, cette horloge étant déjà synchronisée.

Notre premier objectif est de comparer les résultats ainsi obtenus avec ceux présentés auparavant pour notre protocole de synchronisation (V.3.2.1). Nous pouvons ainsi voir sur la Figure V.27 (a) la précision de la synchronisation pour le premier niveau de communication.

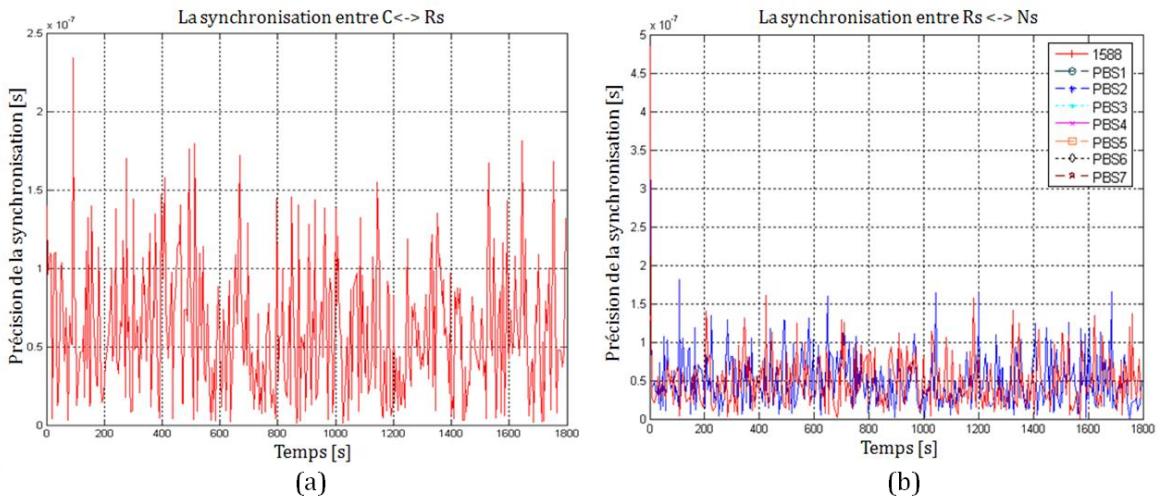


Figure V.27 Synchronization du Niveau 1 et Niveau 2 avec Directionnal-TDMA MAC.

La valeur moyenne obtenue pour la précision de la synchronisation est environ 125ns, ce qui correspond aux performances du protocole IEEE1588-PBS hybride (donné à 150ns).

Pour les paires Routeur-Nœud, la Figure V.27 (b) montre une précision de synchronisation comprise entre 730 ns et 15 ns, comme évalué dans la section V.3.2.1 (5 μ s et 600 ns). Ces résultats démontrent que cette nouvelle configuration (avec des antennes directives) est aussi efficace d'un point de vue performance pour la synchronisation que le protocole lui-même. Mais, cet objectif ne peut être atteint que si le réseau reste performant sur un certain nombre d'indicateurs dont le débit, les délais, la connectivité. Nous venons donc de démontrer que notre modèle d'antennes directives se présente bien de ce point de vue.

EVALUATION DE LA CONSOMMATION D'ENERGIE DU SYSTEME

Comme précisé pour les autres simulations présentées auparavant, pour avoir une évaluation de la consommation électrique proche de réalité, nous utilisons les paramètres usuels d'un Berkeley Mote.

Sachant que le concentrateur est alimenté en permanence par une alimentation continue localisée dans la cabine de l'avion, nous nous intéresserons exclusivement à la consommation des autres éléments du réseau.

Ainsi, si nous analysons le cas des routeurs (Figure V.28), en utilisant une méthode d'extrapolation, nous avons constaté que la durée de vie de la batterie est étendue de 45% dans le cas où nous utilisons des antennes directionnelles.

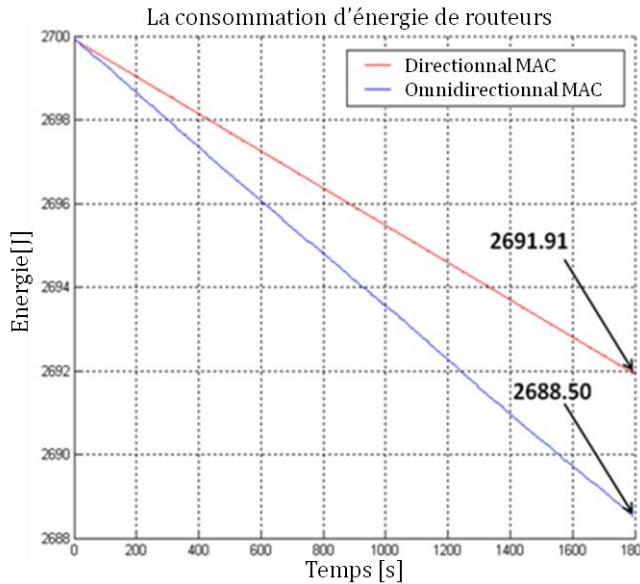


Figure V.28 Consommation en énergie des routeurs.

Comme l'algorithme de synchronisation pour le Niveau 2 de communication requiert deux types de nœuds (nœud 1588 et nœud PBS), nous avons des profils différents de consommation pour chacun. Ainsi, pour un nœud 1588, une augmentation de 25% de la durée de vie est réalisée (Figure V.29).

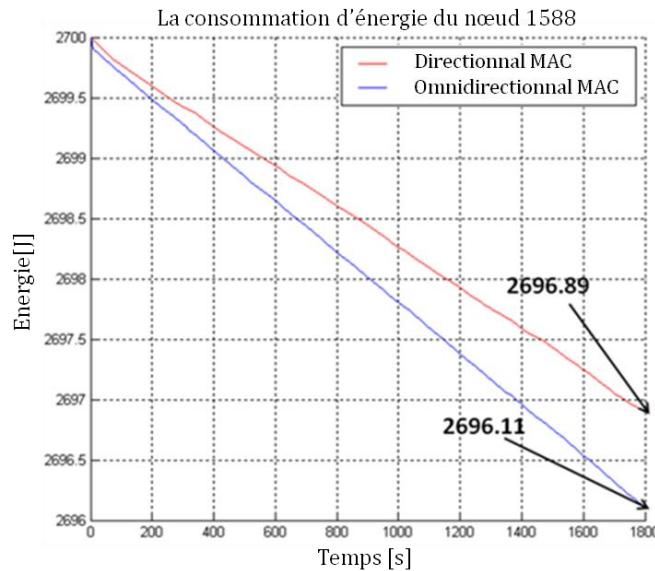


Figure V.29 Consommation en énergie des nœuds 1588.

Pour un nœud PBS (Figure V.30), les simulations ont montré que la durée de vie est alors augmentée de 64%.

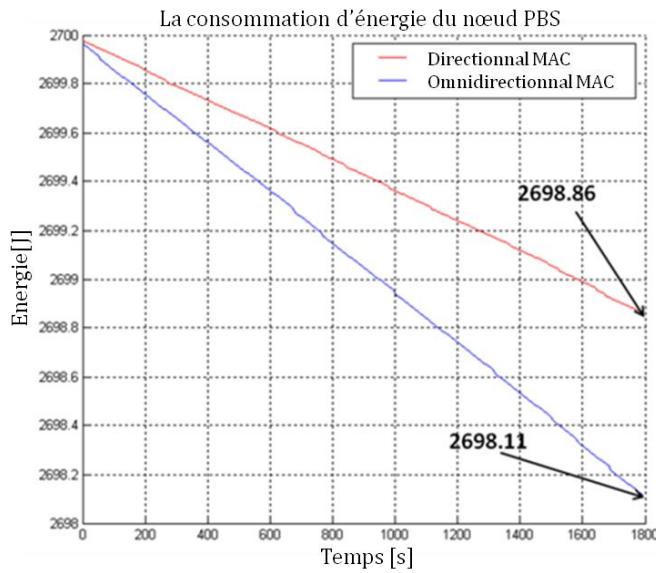


Figure V.30 Consommation en énergie des nœuds PBS.

Ces résultats montrent une différence importante entre les nœuds 1588 et les nœuds PBS. En effet, comme nous l'avons dit plus tôt, un nœud PBS se synchronise seulement en écoutant les communications entre le routeur et le nœud 1588, comme représenté dans la Figure V.31.

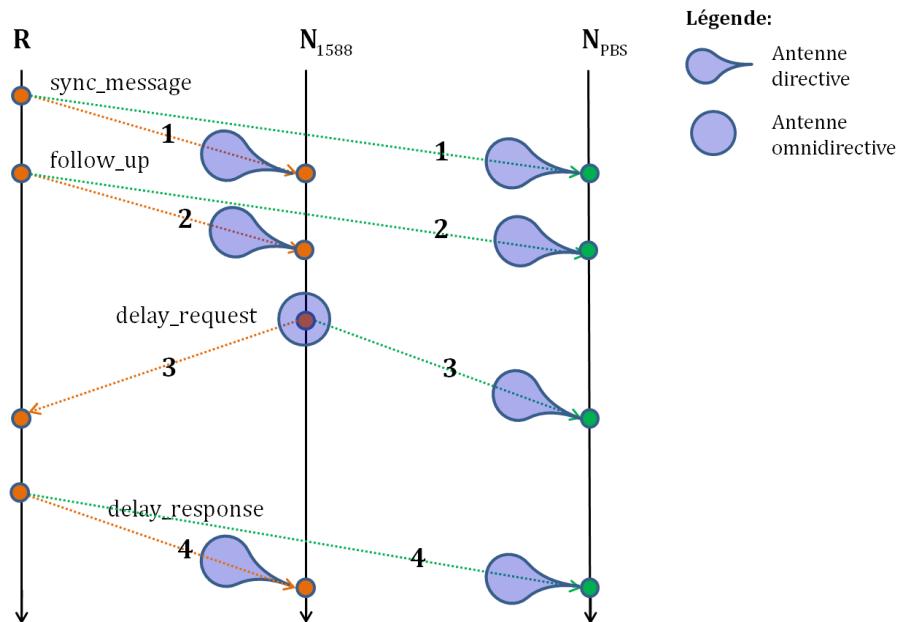


Figure V.31 Utilisation des antennes dans une séquence de synchronisation.

Dans ce cas, le nœud PBS agit comme un récepteur, donc il n'utilise que ses antennes directionnelles et consomme de ce fait moins qu'un nœud 1588 qui doit lui diffuser des informations à l'aide de son antenne omnidirectionnelle.

Dans tous les cas, nos résultats démontrent que l'utilisation des antennes directionnelles dans un réseau de capteurs est très efficace du point de vue économie d'énergie.

V.4.2 Conclusion

Nous avons proposé dans cette section un modèle de réseau de capteurs basé sur des antennes directionnelles, utilisant une méthode de contrôle d'accès au support de type TDMA. Pour cela, une adaptation de différents protocoles a été réalisée dans les couches Physique et Liaison et également dans la modélisation de l'énergie et de sa consommation. Un modèle de simulation de notre architecture a été implémenté sous NS-2 et ce modèle intègre la pile de protocoles proposée pour la sous couche liaison de donnée, Directionnal-TDMA MAC.

Ces simulations ont permis de démontrer la faisabilité et les avantages d'une telle architecture de réseau de capteurs, avantages qui sont les suivants:

- ❖ La précision de la synchronisation, primordiale pour l'application étudiée, s'avère très satisfaisante de l'ordre de quelques dizaines à centaines de nanosecondes;
- ❖ Le système est efficace en termes de couverture, connectivité et délais;
- ❖ Cette nouvelle architecture permet de substantielles économies d'énergie, d'où une augmentation importante de la durée de vie du système.

V.5 **Synthèse générale**

Nous avons présenté au long de ce dernier chapitre les résultats obtenus pendant les étapes de conception et de validations de nos travaux. Ainsi, nous avons traité plusieurs aspects problématiques dans un réseau de capteurs sans fil :

- ❖ Le choix de la topologie réseau, où nous avons proposé une architecture de communication hiérarchique basé sur une méthode d'accès au médium de type TDMA et avec des interfaces multiples;
- ❖ La précision de la synchronisation des données, où nous avons testé et proposé un protocole de synchronisation basé sur la norme IEEE-1588 ;
- ❖ La gestion d'énergie, pour laquelle nous avons proposé un modèle des antennes directives.

Nous rappelons que ce dernier aspect, l'économie d'énergie, a représenté un des critères les plus contraignants, car il a été présent comme objectif aussi dans les deux premières propositions que nous avons faites (le choix de la topologie et la synchronisation).

Les travaux que nous avons accomplis dans ce chapitre peuvent être résumés dans :

- ❖ Une phase d'analyse et conception du système ;
- ❖ Une phase de tests matériels et de mesures ;
- ❖ Une phase de simulation et validation de solutions proposées.

Conclusion Générale et Perspectives

Les réseaux de capteurs sans fil représentent aujourd’hui l’un des systèmes de communication les plus étudiés, en raison de la multitude d’applications qu’il peut satisfaire. Or, cette technologie reste relativement récente (années ‘90), ce qui implique qu’un certain nombre de défis ne soient pas totalement résolus (gestion de la consommation d’énergie, bonne connectivité et couvertures, synchronisation de données de mesures, etc.). Ainsi, les travaux de recherche jouent un rôle important dans l’évolution de ces systèmes, qui sont très différents des types de réseaux classiques, pour développer des techniques et des protocoles adaptés à leurs architectures spécifiques de communication. C’est dans ce cadre que se sont inscrits les travaux de cette thèse.

BILAN

Nous avons donc étudié dans un premier temps les concepts et les techniques existantes pour la mise en œuvre d’un WSN. Cette étude nous a permis d’identifier leurs principales caractéristiques (topologie dynamique, limitation en énergie, mobilité, etc.), qui les éloignent des réseaux de communication traditionnels. En même temps, nous avons conclu que, due à leur capacité de répondre à une multitude des problématiques et applications, il n’est pas possible de proposer une solution de conception générique, qui satisfasse tous les besoins spécifiques.

Nous nous sommes aussi proposés d’étudier les différentes méthodes d’optimisation, qui pourront nous permettre de répondre à des besoins bien particuliers. Ainsi, nous avons conclu qu’une architecture de type cross-layer peut répondre à des problèmes spécifiques, par la création des nouvelles interfaces entre les couches adjacentes ou non-adjacentes qui seront concernées.

Dans ces conditions, et en sachant que le nombre de nœuds capteurs dans un tel réseau est élevé (centaines, milliers des nœuds), nous nous sommes tout d’abord intéressés aux étapes qui précèdent la phase finale de mise en place du système. Ainsi, nous avons étudié les différents outils de simulation, dans le but de choisir celui qui est le plus adapté pour notre application de capteurs sans fil.

A partir de cet état de l’art, nous avons proposé et implanté l’architecture de communication et un certain nombre des solutions optimisées pour l’application concernée.

Nous nous sommes tout d’abord concentrés sur la conception de la topologie réseau d’un système de capteurs sans fil comportant un nombre important d’éléments actifs. L’un des principaux défis dans cette étape a été d’assurer une longue durée de vie du système, ainsi que de maintenir une bonne couverture de détection.

Ainsi, la solution proposée est basée sur une organisation hiérarchique du réseau, structuré sur trois niveaux : le premier est représenté par l’élément central ou le concentrateur, le deuxième par les routeurs et le dernier par les nœuds capteurs. Ensuite, un algorithme pour adapter notre système à un mode de fonctionnement basé sur des interfaces de communication multiples a été conçu.

Pour étudier un tel système, nous avons mis en place un modèle de simulation à travers le logiciel NS-2. Les résultats de simulations montrent que notre solution est efficace par rapport aux différents indicateurs de performance (la synchronisation, les retards, la connectivité, etc.) et réalise également des économies d'énergie de 57% pour les éléments intermédiaires (routeurs) lorsque la configuration des éléments du réseau est optimisée.

Le choix d'un protocole de synchronisation pour un réseau de capteurs sans fil en temps réel (ou un réseau d'instrumentation et de mesure) n'est pas réalisé seulement au regard des caractéristiques et de leurs performances. Dans tous les réseaux sans fil, les problèmes majeurs pour un protocole de synchronisation sont : le décalage des horloges, le temps d'accès au medium, le temps de propagation et le temps d'envoi. La capacité d'éliminer ou de prédire avec précision l'un de ces facteurs, permet d'augmenter considérablement les performances du protocole de synchronisation.

Ainsi, nous avons conclu après une campagne de mesures des performances des cartes NI PCI-1588, que le besoin de la synchronisation pour les applications d'instrumentation et de mesure n'est pas satisfait par les solutions existantes. Ainsi, notre stratégie a été d'adapter le fonctionnement de la norme IEEE-1588, qui offre une très bonne précision de la synchronisation, pour les réseaux de capteurs sans fil.

Par conséquent, nous avons imaginé une architecture de communication transversale, qui contient un nouveau protocole de synchronisation basé sur les principes du protocole PBS et de la norme IEEE-1588, et assure, en plus d'une synchronisation satisfaisante, une importante économie en énergie du système (une prolongation de la durée de vie de la batterie du nœud capteur de 75%).

Dans la deuxième partie de nos travaux, nous avons également proposé un modèle d'antennes directives qui a été intégré dans notre architecture de communication. Nous avons ainsi présenté l'algorithme de fonctionnement et les modules qui ont été implémentés en NS-2. Nous avons montré que les performances globales du système ne sont pas diminuées par rapport à l'utilisation des antennes omnidirectionnelles. De plus, notre modèle d'antennes directives présente un grand avantage en terme d'économie d'énergie (une prolongation de la durée de vie de la batterie d'un routeur de 45% et d'un nœud capteur entre 25% et 64%).

PERSPECTIVES

Ces travaux de thèse ont ouvert un certain nombre de perspectives. Les perspectives de ces travaux de thèse sont les suivantes :

Un des futurs travaux à réaliser par la suite pour l'utilisation des antennes directionnelles est d'analyser le comportement d'un système de communication incluant des éléments mobiles. Même si cela semble bien plus compliqué que dans le cas statique, il est intéressant de noter que le modèle de simulation que nous avons développé inclut déjà cette fonctionnalité. Il reste cependant à trouver comment l'étalonner simplement, étape nécessaire pour obtenir des résultats significatifs et transposables à des systèmes réels.

Il ne faut pas oublier que, même si le plus important est de prolonger au maximum la durée de vie du réseau, le but d'un réseau de capteur est de résoudre le mieux possible le cahier des charges de l'application. Ca veut dire que les autres contraintes comme par exemple la précision de la synchronisation, ne doivent pas être négligées, même si elles sont très consommatrices en énergie. Ainsi, un travail méticuleux doit être réalisé sur la synchronisation des éléments du réseau. La solution sera, en fonction de la stratégie de synchronisation choisie, d'intégrer ce trafic de surcharge dans le trafic des données utiles du réseau.

Une autre perspective envisagée est de concevoir pour notre système un modèle mathématique, par exemple avec des files d'attentes, qui semble adapté à ce contexte. L'idée de cette étape sera de pouvoir exploiter les paramètres de sortie d'un tel modèle, pour pouvoir comparer par la suite avec les résultats de simulations déjà obtenus.

Enfin, il serait nécessaire étant données les applications auxquelles nos travaux s'appliquent d'aborder les thématiques liées à la fiabilité et à la sûreté de fonctionnement du système. En effet, il serait intéressant d'aborder l'impact d'actions malveillantes sur notre réseau. Sans être exhaustif nous pouvons considérer 2 exemples de cas de figure.

Un premier cas de figure pourrait être le suivant. Si les mesures produites sont utilisées par exemple pour la maintenance de l'avion, il est sans doute facile d'empêcher le bon fonctionnement du système ou de le forcer à surconsommer en créant des interférences radio, ce qui réduira sa durée de vie de manière non prévue.

Nous pourrons dans un autre cas « récupérer » les données produites de manière non autorisée puisque nous n'avons pas « protégé » les données transmises. Une méthode initiale d'authentification, et par la suite une technique de cryptage vont de fait augmenter la consommation induite, ce qui constitue un autre axe d'étude possible.

Ces travaux nous ont donc permis d'acquérir une expérience significative dans les Architectures de communication pour les réseaux d'instrumentation sans fil. Et leurs perspectives démontrent clairement que de nouvelles voies de recherche connexes restent finalement ouvertes.

Bibliographie

- [1] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton and J. Zhao, "Habitat monitoring: application driver for wireless communications technology". In Latin America and the Caribbean ACM SIGCOMM Workshop on Data Communications, Costa Rica, Vol. 31, 2001.
- [2] D. Estrin, L. Girod, G. Pottie and M. Srivastava, "Instrumenting the world with wireless sensor networks". International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001), Salt Lake City, UT, Vol. 4, pp. 2033 - 2036, 2001.
- [3] D. Estrin, R. Govindan, J. Heidemann and S. Kumar, "Next century challenges: scalable coordination in sensor networks". Fifth Annual International Conference on Mobile Computing and Networks (MobiCom 1999), Seattle, Washington DC, pp. 263-270, 1999.
- [4] I. F. Akyildiz, W. Su, Y. Sankasubramaniam and E. Cayirci, "Wireless sensor networks: a survey". Computer Networks, Vol. 38, pp. 393-422, 2002.
- [5] <http://business.ezinemark.com/sensor-node-eus-1001-tester-manufacturer-xdectm-ecu-31de241ce9e.html>.
- [6] S. Lee, "An enhanced IEEE 1588 time synchronization algorithm for asymmetric communication link using block burst transmission," Communications Letters, IEEE, Vol. 12, no.9, pp. 687-689, 2008.
- [7] P. Xie and J.-H. Cui, "R-MAC: Aan energy-efficient MAC protocol for underwater sensor networks," Wireless Algorithms, Systems and Applications International Conference, WASA 2007, pp. 187-198, 2007.
- [8] V. Rocha and G. Goncalves, "Sensing the world: challenges on WSNs," 2008 IEEE International Conference on Automation, Quality and Testing, Robotics, Vol. 1, pp. 54-59, 2008.
- [9] Palms fixed/mobile experiment: "Tracking vehicles with a UAV-delivered sensor network", Marine Corps Air/Ground Combat Center, UC Berkeley and MLB Co, California.
- [10] C. E. Perkins, "Ad hoc networking. Addison-Wesley", Addison-Wesley Longman Publishing Co., Inc. Boston, USA, ISBN: 0-201-30976-9, 2001.
- [11] I. Stojmenovic, "Handbook of wireless networks and mobile computing", Wiley, ISBN: 978-0-471-41902-0, 2002.
- [12] N. P. Mahalik, "Fieldbus Technology – Industrial network standards for real-time distributed control", Springer-Verlag Berlin, Germany, ISBN 3-540-40183-0, 2003.
- [13] A. Willig, "Wireless LAN technology for the factory floor: challenges and approaches"; Citation information in R. Zurawski, "Handbook on Industrial Communication Systèmes", CRC Press, ISBN: 978-0-8493-1985-3, 2005.
- [14] M. Sgroi, A. Wolisz, A. Sangiovanni-Vincentelli and J. M. Rabaey, "A service-based universal application interface for ad-hoc wireless sensor networks", BWRC Winter 2004 Retreat, 2004.
- [15] W. Ye, J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless Sensor networks", Proceedings of INFOCOM 2002, IEEE Press, New York, pp. 1567-1576, 2002.
- [16] S. Adireddy and L. Tong, "Medium Access Control with channel state information for large sensor networks", IEEE International Workshop on Multimedia Signal Processing, St. Thomas, Virgin Islands, pp. 416 – 419, 2002.

- [17] A. Woo and D. Culler, "A transmission control scheme for media access in sensor networks", *Seventh Annual International Conference on Mobile Computing and Networking 2001 (MobiCom)*, Rome, Italy, pp. 221—235, 2001.
- [18] P. H. Lehne and M. Pettersen, "An overview of smart antenna technology for mobile communications systems", *IEEE Communications Surveys and Tutorials*, Vol. 2, no. 4, pp. 2-13, 1999.
- [19] I. Cidon and M. Sidi, "Distributed assignment algorithms for multihop packet radio networks", *IEEE Transactions on Computers*, Vol. 38, no. 10, pp. 1353-1361, 1989.
- [20] LAN/MAN Standards Committee of the IEEE Computer Society. *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)*, 2003.
- [21] J. Beutel, O. Kasten, F. Mattern, K. Romer, F. Siegemund and L. Thiele, "Prototyping wireless sensor network applications with BTNodes", *First European Workshop Wireless Sensor Networks (EWSN 2004)*, Berlin, Germany, Vol. 2920, pp. 323-338, 2004.
- [22] F. Babich and M. Comisso, "Throughput and delay analysis of 802.11-based wireless networks using smart and directional antennas," *Journal IEEE Communications and Transactions*, Vol. 57, no. 5, pp. 1413-1423, 2009.
- [23] J. C. Haartsen, "The Bluetooth radio système", *IEEE Personal Communications*, Vol. 7, no. 1, pp. 28-36, 2000.
- [24] T. V. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks", *1st ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, pp. 171-180, 2003.
- [25] J. Polastre, J. Hill and D. Culler, "Versatile low power media access for wireless sensor networks", *2nd ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*, Baltimore, MD, pp. 95-107, 2004.
- [26] N. Gajaweera and D. Dias, "FAMA/TDMA hybrid MAC for wireless sensor networks", *4th International Conference on Information and Automation for Sustainability, ICIAFS 2008*, pp. 67-72, 2008.
- [27] C. Rohl, H. Woesner and A. Wolisz, "A short look on power saving mechanisms in the wireless LAN standard IEEE 802.11", In J. M. Holtzmann and M. Zorzi, "Advances in Wireless Communications", Vol. 435, no. 3, pp. 219-226, Kluwer Academic Publishers, 1998.
- [28] V. Kawadia and P.R. Kumar, "A cautionary perspective on cross-layer design", *IEEE Wireless Communications*, Vol. 12, no. 1, pp. 3- 11, 2005.
- [29] A. J. Goldsmith and S. B. Wicker, "Design challenges for energy-constrained ad hoc wireless networks", *IEEE Wireless Communications*, Vol. 9, pp. 8-27, 2002.
- [30] V. Handziski, H. Karl, A. Kopke and A. Wolisz, "A common wireless sensor network architecture", *Proceedings 1. GI/ITG Fachgespräch "Sensornetze"*. Technical Report TKN-03- 012 of the Telecommunications Networks Group, Technische Universität Berlin, Germany, pp. 10-17, 2003.
- [31] A. Willig, M. Kubisch, C. Hoene and A. Wolisz, "Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer". *IEEE Transactions on Industrial Electronics*, Vol. 49, no. 6, pp. 1265-1282, 2002.
- [32] S. Roundy, D. Steingart, L. Frechette, P. Wright and J. Rabaey, "Power sources for wireless sensor networks", *1st European Workshop on Wireless Sensor Networks (EWSN)*. LNCS, Springer, Berlin, Germany, Vol. 2920, pp. 1-17, 2004.

- [33] L. Benini and G. De Micheli, "Dynamic power management: design techniques and CAD tools". Kluwer, ISBN 978-0-7923-8086-3, 1998.
- [34] T. S. Rappaport. "Wireless Communications – Principles and practice", Prentice Hall, Upper Saddle River, NJ, 2002.
- [35] G. J. Pottie and W. J. Kaiser, "Embedding the internet: wireless integrated network sensors", Communications of the ACM, Vol. 43, no. 5, pp. 51–58, 2000.
- [36] J. Hill and D. Culler, "MICA: A Wireless Platform for Deeply Embedded Networks", IEEE Micro, Vol. 22, no. 6, pp. 12–24, 2002.
- [37] A. Sinha, A. Chandrakasan, "Dynamic power management in wireless sensor networks", IEEE Design and Test of Computers, Vol. 18, pp. 62 – 74, 2001.
- [38] J.-P. Ebert and A. Wolisz, "Combined tuning of RF power and medium access control for WLANs", MONET - Mobile Networks and Applications, Vol.6, no. 5, pp. 417–426, 2000.
- [39] B. Narendran, J. Sienicki, S. Yajnik and P. Agrawal, "Evaluation of an adaptive power and error control algorithm for wireless Systems", International Conference on Communication (ICC), Montreal, Canada, Vol. 1, pp. 349-355, June 1997.
- [40] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, " Energy-efficient communication protocol for wireless micro sensor networks", Hawaii International Conference on Systems Science, Vol. 8, pp. 8020, 2000.
- [41] B. Radunovic and J.-Y. Le Boudec, "Optimal power control, scheduling and routing in UWB networks", IEEE Journal on Selected Areas in Communications, Vol. 22, no. 7, pp. 1252-1270, 2004,
- [42] R. Merz, J.-Y. Le Boudec, J. Widmer and B. Radunovic, "A rate-adaptive MAC protocol for low-power ultra-wide band ad-hoc networks", 3rd International Conference on AD-HOC Networks and Wireless (Ad-Hoc Now), Vancouver, British Columbia, Canada, Vol. 3158, pp. 306-311, 2004.
- [43] J. Elson and D. Estrin, "Time synchronization for wireless sensor networks", 15th International Parallel and Distributed Processing Symposium (IPDPS-01), IEEE Computer Society, 2001.
- [44] Z. Tian, X. Luo and G.B. Giannakis, "Cross-layer sensor network synchronization", Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, Vol. 1, pp. 1276 – 1280, 2004.
- [45] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks", Ad Hoc Networks, Vol. 3, pp. 325–349, 2005.
- [46] V. Kawadia and P. R. Kumar, "Power control and clustering in ad hoc networks", IEEE INFOCOM, San Francisco, CA, Vol. 1, pp. 459-469, 2003.
- [47] I. Demirkol, C. Ersoy and F. Alagoz, "MAC protocols for wireless sensor networks: a survey," IEEE Communications Magazine, Vol. 44, no.4, pp. 115- 121, 2006.
- [48] B. Sundararaman, U. Buy and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey" Journal Ad Hoc Networks, Vol. 3, pp. 281-323, 2005.
- [49] D. L. Mills, "Network Time Protocol", Version 3: Specification, Implementation and Analysis, Network Working Group, University of Delaware, 1992.
- [50] D. L. Mills. "Internet time synchronization: the network time protocol", IEEE transactions on communications, Vol. 39, no. 10, pp. 1482–1493, 1991.
- [51] A.R. Swain and R.C. Hansdah, "An energy efficient and fault-tolerant clock synchronization protocol for wireless sensor networks", COMSNETS, pp. 1-10, 2010.
- [52] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systèmes/ IEEE, New York, 2002.

- [53] H. Cho, J. Jung, B. Cho, Y. Jin, S.-W. Lee and Y. Baek, "Precision time synchronization using IEEE 1588 for wireless sensor networks", International Conference on Computational Science and Engineering, Vol. 2, pp. 579-586, 2009.
- [54] J. Elson, L. Girod and D. Estrin, "Fine-grained network time synchronization using reference broadcasts", Proc. Fifth Symp. Operating Systèmes Design and Implementation (OSDI '02), Vol. 36, pp. 147 – 163, 2002.
- [55] A. Hu and S.D. Servetto, "Algorithmic aspects of the time synchronization problem in large-scale sensor networks," ACM/Kluwer J. Mobile Networks and Applications, special issue on wireless sensor networks, Vol. 10, pp. 491 – 503, 2003.
- [56] M. Maroti, B. Kusy, G. Simon and A. Ledeczi, "The flooding time synchronization protocol", Second ACM Conf. Embedded Networked Sensor Systems (SenSys '04), pp. 39-49, 2004.
- [57] K.-L. Noh, E. Serpedin and K. Qaraqe, "A new approach for time synchronization in wireless sensor networks: Pairwise Broadcast Synchronization", IEEE Transaction on Wireless Communication, Vol. 7, no. 9, pp. 3318-3322, 2008.
- [58] M. Maroti, B. Kusy, G. Simon and Á. Lédeczi, "Robust multi-hop time synchronization in sensor networks", International Conference on Wireless Networks, pp. 454-460, 2004.
- [59] D. Macii, A. Ageev and A. Somov, "Power consumption reduction in wireless sensor networks through optimal synchronization", I2MTC, pp. 1346-1351, 2009.
- [60] S. H. Lee and L. Choi, "Chaining clock synchronization: an energy-efficient clock synchronization scheme for wireless sensor networks", 10th International Symposium on Pervasive Systèmes, Algorithms, and Networks, pp. 172-177, 2009.
- [61] K.-L. Noh, Q. Chaudhari, E. Serpedin and B. Sutert, "Power-efficient clock synchronization using tow-way timing message exchange in wireless sensor networks", Military Communications Conference (MILCOM), pp. 1-7, 2006.
- [62] Y. Sun, J. Qian and J. Wu, "Hybrid energy-aware time synchronization protocol for WSNs in coal mine", International Conference on Information Acquisition, pp. 436-441, 2007.
- [63] G. J. Pottie and W. J. Kaiser, "Embedding the internet: wireless integrated network sensors", Communications of the ACM 2000, Vol. 43, no. 5, pp. 51–58, 2000.
- [64] H. Karl and A. Willing, "Protocols and architectures for wireless sensor networks", ISBN: 0-470-09510-5, pp. 139-145, pp. 251-286, pp 289-329, 2005.
- [65] J. Foerster, E. Green, S. Somayazulu and D. Leeper, "Ultra-Wide Band technology for short or medium range wireless communications", Intel Architecture Labs, Intel Technology Journal Q2, 2001.
- [66] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for ad hoc sensor networks", IEEE Transaction on Mobile computing, Vol. 3, no. 4, pp. 366-379, 2004.
- [67] R. Rajaraman, "Topology control and routing in ad hoc networks: a survey", ACM SIGACT News, Vol. 33, no. 2, pp. 60–73, 2002.
- [68] C. Chiasserini, I. Chlamtac, P. Monti and A. Nucci, "Energy-efficient design of wireless ad hoc networks", IFIP Networking, Pisa, Italy, Vol. 2345, pp. 376-386, 2002.
- [69] T. J. Kwon and M. Gerla, "Clustering with power control", MILCOM, Atlantic City, NJ, Vol. 2, pp. 1424–1428, 1999.
- [70] P. Levis, N. Lee, M. Welsh and D. Culler, "Tossim: accurate and scalable simulation of entire TinyOS applications", 1st international conference on Embedded networked sensor systems, New York, USA, pp. 126–137, 2003.

- [71] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen and M. Welsh, "Simulating the power consumption of large-scale sensor network applications". Conference on Embedded Networked Sensor Systems (SynSys), ACM Press, pp. 188–200, 2004.
- [72] A. Fraboulet, G. Chelius and E. Fleury, "Worldsens: development and prototyping tools for application specific wireless sensors networks", in IPSN'07 Track on Sensor Platforms, Tools and Design Methods, Cambridge, Massachusetts, USA, 2007.
- [73] http://www.bores.com/courses/intro/chips/6_basics.htm.
- [74] S.-F. Li, R. Sutton and J. Rabaey, "Low power operating système for heterogeneous wireless communication systems", 10th International Conference on Parallel Architectures and Compilation Techniques (PACT 01), Barcelona, Spain, pp. 1-16, 2001.
- [75] D. Cavin, Y. Sasson and S A. chiper, « On the accuracy of MANET simulators», Second ACM international workshop on Principles of mobile computing, pp. 38–43, 2002.
- [76] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler and K. S. J. Pister, "Système architecture directions for networked sensors", 9th International Conference on Architectural Support for Programming Languages and Operating Systèmes, Cambridge, MA, pp. 93–104, 2000.
- [77] B. Hofmann-Wellenhof, H. Lichtenegger and J. Collins, "Global Positioning System: theory and practice", Springer, 4th edition, ISBN 978-3-211-83534-0, 1997.
- [78] BTnodes: A Distributed Environment for Prototyping Ad Hoc Networks, October 2004. Website <http://www.btnode.ethz.ch>.
- [79] H. Ritter. ScatterWeb. Website <http://www.scatterweb.de/>, 2004.
- [80] M. Srivastava, R. Muntz and M. Potkonjak, "Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments" (Challenge Paper). 7th International Conference on Mobile Computing and Networking, ACM Press, Rome, Italy, pp. 132–138, 2001.
- [81] L. Schwiebert, S. K. S. Gupta and J. Weinmann, "Research challenges in wireless networks of biomedical sensors", 7th International Conference on Mobile Computing and Networking (ACM Mobicom), Rome, Italy, pp. 151–165, 2001.
- [82] ETSI. TR 101 683, HIPERLAN Type 2: Systems Overview. ETSI, 2000.
- [83] G. T. Huang, "Casting the Wireless Sensor Net", Technology Review, pp. 51–56, 2003.
- [84] P. Bonnet, J. E. Gehrke and P. Seshadri, "Querying the physical world", IEEE Personal Communications, Vol. 7, no. 5, pp. 10–15, 2000.
- [85] J. M. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel and S. Roundy, "PicoRadio supports ad hoc ultra-low power wireless networking", IEEE Computer, Vol. 33, no. 7, pp. 42–48, 2000.
- [86] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, "Wireless sensor networks for habitat monitoring", 1st ACM Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, pp. 88-97, 2002.
- [87] J. Kulik, W. R. Heinzelman and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," Wireless Networks, Vol. 8, pp. 169–185, 2002.
- [88] B. You, G. Chen and W. Guo, "Topology control in wireless sensor networks based on discrete particle swarm optimization", IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2009, Vol. 1, pp.269-273, 2009.
- [89] H.J. Song, M.H. Zhu and Y.T. Bai, "Design of general purpose simulation package of SAR system," Radar 97 (Conf. Publ. No. 449), pp. 697-699, 1997.

- [90] C.-Y. Wan, A.T. Campbell and L. Krishnamurthy, "Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks", IEEE Journal on Selected Areas in Communications, Vol.23, no.4, pp. 862- 872, 2005.
- [91] A. El-Hoiydi, "Spatial TDMA and CSMA with preamble sampling for low power ad hoc wireless sensor networks," 7th IEEE International Symposium on Computers and Communications (ISCC'02), pp. 685-692, 2002.
- [92] L. Kleinrock and F. Tobagi, "Packet switching in radio channels, Part I: Carrier Sense Multiple Access Modes and their throughput delay characteristics," IEEE Transactions on Communications, Vol. 23, no. 12, pp. 1400-1416, 1975.
- [93] O. Sharon and E. Altman, "An efficient polling MAC for wireless LANs", IEEE/ACM Transactions on Networking, Vol. 9, no. 4, pp. 439-451, 2001.
- [94] S. Ganeriwal, R. Kumar and M. B. Srivastava, "Timing-Sync protocol for sensor networks", 1st ACM Conference on Embedded Networked Sensor Systems (SenSys'03), Los Angeles, pp. 138-149, 2003.
- [95] J.-J. Chen and C.-F. Kuo, "Energy-efficient scheduling for real-time systems on Dynamic Voltage Scaling (DVS) platforms", 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA 2007, pp. 28-38, 2007.
- [96] M. Walchli, R. Zurbuchen, T. Staub and T. Braun, "Gravity-based local clock synchronization in wireless sensor networks", NETWORKING 2009, pp. 907-918, 2009.
- [97] D. Trezentos, "Standard pour réseaux sans fil: IEEE 802.11", Techniques de l'ingénieur. Sécurité des systèmes d'information, Vol. 1, pp. 1-12, 2002.
- [98] S. Tabbane, "Réseaux mobiles. Réseaux et télécommunications", Hermès, Paris, France, 1997.
- [99] J. Henaut, D. Dragomirescu, F. Perget and R. Plana, "Validation of the MB-OFDM modulation for high data rate WSN for satellite ground testing," Fifth International Conference on Systems (ICONS), pp. 41-46, 2010.
- [100] P.-A. Muller and N. Gaertner, "Modélisation objet avec UML", Editions eyrolles, ISBN : 2212091222, 2001.
- [101] P. Roques and F. Vallée, "UML en action", Groupe Eyrolles, ISBN : 2-212-11213-0, 2003.
- [102] R. Ma, L. Xing, T. Jin and T. Song, "A data transmission mechanism for survivable sensor networks," IEEE International Conference on Networking, Architecture, and Storage, NAS 200, pp. 9-15, 2009.
- [103] S. U. Donapudi, C.O. Obel and J. Madsen, "Extending lifetime of wireless sensor networks using Forward Error Correction," 24th Norchip Conference, pp. 277-280, 2006.
- [104] X. Leiming, CSCW Lab. at CS Dept, "How to add a new protocol in NS2", Tsinghua Univ., 2001.
- [105] K. Fall and K. Varadhan, "The ns manual", A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC. (The VINT Project), 2010.
- [106] D.C. Jefferson, S.M. Lichten, L.E. Young, "A test of precision GPS clock synchronization," Proceedings of the 1996 IEEE International 50th Frequency Control Symposium, pp. 1206-1210, 5-7 Jun 1996.
- [107] Y. Bang, J. Han, K. Lee, J. Yoon, J. Joung, S. Yang and J.-K.K. Rhee, "Wireless network synchronization for multichannel multimedia services," 11th International Conference on Advanced Communication Technology, ICACT 2009, Vol. 2, pp. 1073-1077, 2009.
- [108] T. Cooklev, J.C. Eidson and A. Pakdaman, "An implementation of IEEE 1588 over IEEE 802.11b for synchronization of wireless local area network nodes," IEEE

- Transactions on Instrumentation and Measurement, Vol. 56, no. 5, pp. 1632-1639, 2007.
- [109] J. Kannisto, T. Vanhatupa, M. Hannikainen and T.D. Hamalainen, "Software and hardware prototypes of the IEEE 1588 precision time protocol on wireless LAN," 14th IEEE Workshop on Local and Metropolitan Area Networks, LANMAN 2005, pp. 6, 2005.
- [110] H. Abubakari and S. Sastry, "IEEE 1588 style synchronization over wireless link," IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, ISPCS 2008, pp. 127-130, 2008.
- [111] L. De Vito, S. Rapuano, L. Tomaciello, "One-Way Delay Measurement: State of the Art", IEEE Transactions on Instrumentation and Measurement, Vol.57, no. 12, pp. 2742-2750, Dec. 2008.
- [112] R. Brault and R. Piat, "Les antennes", Editions Techniques et Scientifiques Françaises, ISBN 2-10-023993-7, 2001.
- [113] A. Ranjitkar and Y.-B. Ko, "A collaborative routing for wireless multi-hop networks with directional antennas", Second International Conference on Ubiquitous and Future Networks (ICUFN), pp. 127-132, 2010.
- [114] K. B. Johnsson and D. C. Cox, "An adaptive cross-layer scheduler for improved QoS support of multiclass data services on wireless systems", IEEE Journal on Selected Areas in Communications, Vol. 23, no. 2, pp. 334-343, 2005.
- [115] R. R. Choudhury, X. Yang, R. Ramanathan and N. Vaidya, "Using directional antennas for medium acces control in ad hoc networks", MobiHoc'02, pp. 59-70, 2002.
- [116] T. Korakis, G. Jakllari and L. Tassiulas, "A MAC protocol for full exploitation of directional antennas in AD-hoc Wirless networks", MobiHoc'03, pp. 98-107, 2003.
- [117] M. Takai, J. Martni, A. Ren and R. Bagrodia, "Directionnal virtual carrier sensing for directional antennas in mobile Ad hoc networks", MobiHoc'02, pp. 183-193, 2002.
- [118] S. Kandasamy, R. Campos, R. Morla and M. Ricardo, "Using directional antennas on stub wireless mesh networks: impact on throughput, delay, and fairness", International Conference on Computer Communications and Networks (ICCCN), pp. 1-6, 2010.
- [119] S. Basagni, "Distributed clustering for ad hoc networks," I-SPAN'99, pp. 310-315, 1999.
- [120] A. Lecointre, D. Dragomirecu, P. Berthou and R. Plana, "A reconfigurable IR-UWB radio interface, with directional antennas and localization capability", 14th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2007, pp. 1055 -1058, 2007.
- [121] K.-T. Feng, "LMA: Location- and Mobility-Aware medium-access control protocols for vehicular ad hoc networks using directional antennas," IEEE Transactions on Vehicular Technology, Vol. 56, no. 6, pp. 3324-3336, 2007.
- [122] X. Xie, B. Huang, S. Yang and T. Lv, "Adaptive multi-channel MAC protocol for dense VANET with directional antennas", 6th IEEE Consumer Communications and Networking Conference, CCNC 2009, pp. 1-5, 10-13, 2009.
- [123] A. Akram and M. Shafqat, "Battery and frequency optimized AODV for wireless mesh networks", Canadian Journal on Multimedia and Wireless Networks, Vol. 1, no. 3, pp.31-37, 2010.
- [124] S. Ahmad, I. Awan, A. Waqqas and Ahmad B., "Performance Analysis of DSR & Extended DSR Protocols", Second Asia International Conference on Modeling & Simulation, AICMS 08, pp. 191-196, 2008.

- [125] Q. Liu, H. Wang, J. Kuang, Z. Wang and Z. Bi, "WSNp1-1: M-TORA: a TORA-based multi-path routing algorithm for mobile ad hoc networks", IEEE Global Telecommunications Conference, GLOBECOM'06, pp. 1-5, 2006.
- [126] C.-K. Toh, "Long-lived ad hoc routing based on the concept of associativity", Mobile Multimedia & HiSpeed Networking Laboratory, IETF Draft, 8 pages, 1999.
- [127] R. Dube, C. D. Rais, K.-Y. Wang and S. K. Tripathi, "Signal stability based adaptive routing for ad hoc mobile networks", IEEE Pers. Comm., pp. 36-45, 1997.
- [128] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers", SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications, pp. 234-244, 1994.
- [129] S. Murthy and J.J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks", ACM Mobile Networks and Applications in Mobile Communication Networks, pp. 183-97, 1996.
- [130] T. Clausen, "Mobile ad hoc networking (MANET)" Internet-Draft LIX, Ecole Polytechnique, France, 2005.
- [131] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla and T.-W. Chen, "Scalable routing strategies for ad hoc wireless networks" IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks, pp. 1369-79, 1999.
- [132] W. Si, A. Zomaya and S. Selvadurai, "A geometric deployment and routing scheme for directional wireless mesh networks", IEEE Transactions on Computers, 2010.
- [133] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks", Int. Conf. on Systems Science, pp. 3005-3014, 2000.
- [134] A. Manjeshwar and D.P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks", 15th International in Parallel and Distributed Processing Symposium, pp. 2009-2015, 2001.
- [135] D. Xiao, M. Wei and Y. Zhou, "Secure-SPIN: Secure Sensor Protocol for Information via Negotiation for wireless sensor networks", 1ST IEEE Conference on Industrial Electronics and Applications, pp. 1-4, 24-26 2006.
- [136] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva, "Directed diffusion for wireless sensor networking", Transactions on Networking, IEEE/ACM , Vol. 11, no. 1, pp. 2- 16, 2003.
- [137] I.K. Maarouf and T.R. Sheltami, "Simulation-based analysis for GEAR performance optimization", 22nd International Conference on Advanced Information Networking and Applications - Workshops, AINAW 2008, pp. 910-915, 25-28, 2008.
- [138] M. Takai, J. Zhou and R. Bagrodia, "Adaptive range control using directional antennas in mobile ad hoc networks", In Proceedings of MSWiM'2003, pp. 92-99, 2003.
- [139] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks", WSNA'02, pp. 32-41, 2002.
- [140] H. Zhang and J. Hou, "Maintaining sensing coverage and connectivity in large sensor networks", Ad Hoc & Sensor Wireless Networks, Vol. 1, no. 1-2, pp. 89-124, 2005.
- [141] H. Elham, J. Kamal and B. Ali, "Increasing WSN lifetime by using learning automata for optimal route selection", ICINA 2010, pp. 215-218, 2010.
- [142] C.H. Wang and S.K. Pavuluri, "Fabrication and assembly of high gain MEMS antennas for wireless communications", ECTC 2008, pp. 1941-194, 2008.
- [143] C.A. Balanis, "Modern antenna handbook", John Willey & sons, ISBN: 0470036346, 2008.
- [144] C.A. Balanis, "Antenna theory: analysis and design, 3rd Edition", John Willey & sons, ISBN: 978-0-471-66782-7, 2005.

- [145] M. Klemm, I.Z. Kovacs, G.F. Pedersen and G. Troster, "Comparison of directional and omni-directional UWB antennas for Wireless body area network applications", ICECom 2005, pp. 1-4, 2005.
- [146] E. Albu, Y. Labit, T. Gayraud and P. Berthou, "An energy-efficient clock synchronization protocol for wireless sensor networks", IFIP Wireless Days 2010, Italy, pp. 5, 2010.
- [147] H. Cho, J. Jung, B. Cho, Y. Jin, S.-W. Lee and Y. Baek, "Precision Time Synchronization using IEEE 1588 for Wireless Sensor Networks", International Conference on Computational Science and Engineering, Vol. 2, pp. 579-586, 2009.
- [148] S. Lee, "An enhanced IEEE 1588 time synchronization algorithm for asymmetric communication link using block burst transmission", IEEE Communications Letters, Vol. 12, no. 9, pp. 687 – 689, 2008.
- [149] R.L. Scheiterer, N. Chongning, D. Obradovic, G. Steindl and F.-J. Goetz, "Synchronization performance of the Precision Time Protocol in the face of slave clock frequency drift", IEEE International Conference on Automation Science and Engineering, Vol. 23-26, pp. 554 – 559, 2008.
- [150] B. Baynat, "Théorie des fils d'attente. Des chaînes de Markov aux réseaux à forme produit », Hermès-Lavoisier, ISBN 2-7462-0120-8, pp. 29, 2000.

Table des figures

| | | |
|--------------|---|----|
| Figure I.1 | Exemple d'architecture de réseau de capteurs embarqués sans-fil..... | 9 |
| Figure I.2 | Les principaux composants matériels d'un nœud capteur..... | 13 |
| Figure I.3 | Nœud capteur Mica2 Mote..... | 15 |
| Figure I.4 | Nœud capteur du projet EYES..... | 15 |
| Figure I.5 | Nœud capteur de la plate-forme ScatterWeb..... | 15 |
| Figure I.6 | Nœud capteur BTnode..... | 15 |
| Figure I.7 | Modèle de la pile protocolaire pour WSNs..... | 16 |
| Figure I.8 | Réseau MANET..... | 19 |
| Figure I.9 | Exemple de réseau bus de terrain..... | 20 |
| Figure I.10 | Topologies LR-WPAN..... | 21 |
| Figure I.11 | Réseau scatternet Bluetooth..... | 22 |
| Figure I.12 | Différents dispositif de récupération d'énergie..... | 26 |
| Figure I.13 | Exemple d'architectures cross-layer générique..... | 28 |
| Figure I.14 | Différentes options de contrôle de topologie [64]..... | 30 |
| Figure I.15 | L'architecture de base pour le simulateur NS-2..... | 43 |
| Figure I.16 | Réseau de capteurs embarqués sans fil..... | 45 |
| Figure I.17 | Topologie du réseau..... | 45 |
| Figure I.18 | Techniques pour minimiser la consommation d'énergie..... | 48 |
| Figure II.1 | Les flux de communication dans la topologie SACER..... | 55 |
| Figure II.2 | Les acteurs de notre système..... | 59 |
| Figure II.3 | Le diagramme du cas d'utilisation relatif au concentrateur..... | 60 |
| Figure II.4 | Le diagramme du cas d'utilisation relatif au routeur..... | 61 |
| Figure II.5 | Diagramme du cas d'utilisation relatif au nœud..... | 62 |
| Figure II.6 | Les éléments du processus de communication..... | 64 |
| Figure II.7 | Topologie de niveau MAC Concentrateur-Routeur-Nœud..... | 65 |
| Figure II.8 | Organisation d'une super-trame de la liaison de données..... | 65 |
| Figure II.9 | Format du PDU de la liaison de données..... | 66 |
| Figure II.10 | Diagramme de séquence d'échange des trames d'un cycle normal..... | 66 |
| Figure II.11 | PDU de Niveau 2 et son champ de correction d'erreur..... | 67 |
| Figure II.12 | Adaptation de la norme OFDM ECMA-368 au système OFDM SACER..... | 67 |
| Figure II.13 | Topologie logique du réseau SACER..... | 68 |
| Figure II.14 | PDU réseau descendant..... | 68 |
| Figure II.15 | PDU réseau ascendant..... | 68 |
| Figure II.16 | Message applicatif de relevé de mesure..... | 69 |
| Figure II.17 | Message applicatif de configuration..... | 69 |
| Figure II.18 | Diagramme d'échange des messages pour la Découverte du réseau..... | 70 |
| Figure II.19 | Format d'un message applicatif de découverte..... | 70 |
| Figure II.20 | Format d'un message applicatif de réglage d'un paramètre capteur..... | 70 |
| Figure II.21 | La trame de Bannissement..... | 71 |
| Figure II.22 | Diagramme de séquence d'échange des trames du Bannissement..... | 71 |
| Figure II.23 | L'ensemble des trames pouvant circuler dans le système..... | 72 |
| Figure II.24 | Carte Wisair Wireless USB PCI Express Mini..... | 72 |
| Figure II.25 | Carte NI PCI-1588..... | 74 |
| Figure II.26 | La plateforme ML410 FPGA-Xilinx..... | 75 |
| Figure II.27 | Trafic de données utiles dans notre réseau hiérarchique..... | 79 |
| Figure II.30 | Organisation du réseau avec des interfaces multiples (Solution I)..... | 80 |
| Figure II.31 | Organisation du réseau avec des interfaces multiples (Solution II)..... | 81 |
| Figure III.1 | La première phase de synchronisation de l'IEEE-1588..... | 87 |
| Figure III.2 | La deuxième phase de synchronisation de l'IEEE-1588..... | 87 |
| Figure III.3 | Principe de synchronisation du protocole PBS..... | 88 |
| Figure III.4 | L'échange de messages pour le protocole PBS..... | 89 |
| Figure III.5 | Schéma de synchronisation..... | 91 |
| Figure III.6 | L'échange des messages dans l'IEEE1588-PBS hybride..... | 91 |

| | | |
|----------------------|---|-----|
| Figure III.7 | L'architecture hiérarchique du réseau par aile | 93 |
| Figure III.8 | Les niveaux de synchronisation dans le réseau..... | 93 |
| Figure III.9 | Les messages échangés dans le Niveau 2 de synchronisation..... | 94 |
| Figure III.10 | Intégration de l'IEEE-1588 dans l'architecture du logiciel NS-2..... | 98 |
| Figure III.11 | Intégration du l'IEEE1588-PBS hybride dans l'architecture du logiciel NS-2.... | 100 |
| Figure V.1 | La topologie réseau de notre système..... | 120 |
| Figure V.2 | Le temps d'un cycle de communication (cycle normal de données)..... | 121 |
| Figure V.3 | Comparaison sur les trois cas étudiés..... | 121 |
| Figure V.4 | L'échange des messages dans le protocole IEEE1588-PBS hybride. | 123 |
| Figure V.5 | Deux scénarios d'organisations du réseau avec des interfaces multiples..... | 124 |
| Figure V.6 | La synchronisation pour le Niveau1 et le Niveau2 de communication | 125 |
| Figure V.7 | Durée cycle de transmission entre C-Routeurs et R-Nœuds. | 125 |
| Figure V.8 | La consommation d'énergie pour les nœuds capteurs pour les routeurs..... | 126 |
| Figure V.9 | Plate-forme d'expérimentation..... | 129 |
| Figure V.10 | Fonctionnement d'IEEE-1588 | 129 |
| Figure V.11 | Capture des signaux PFI sur l'oscilloscope (en filaire). | 130 |
| Figure V.12 | Signaux en sortie des PFIs..... | 131 |
| Figure V.13 | Mesures de la variation de la gigue en connexion filaire (une seule capture).. | 132 |
| Figure V.14 | Analyse de performances pour la variation de la gigue (~55 captures)..... | 132 |
| Figure V.15 | Capture des signaux PFI sur l'oscilloscope (en sans fil)..... | 133 |
| Figure V.16 | Mesures de la variation de la gigue en sans fil (deux méthodes, une capture). .. | 133 |
| Figure V.17 | Analyse de performances pour la variation de la gigue (~55 captures)..... | 134 |
| Figure V.18 | Signaux et la variation de la gigue pour une connexion filaire (méthode 1). | 135 |
| Figure V.19 | Signaux et la variation de la gigue pour une connexion sans fil (méthode 1) .. | 135 |
| Figure V.20 | Plate-forme de test filaire et sans fil. | 136 |
| Figure V.21 | Délai entre Maitre-Esclave et Esclave-Maitre sur une liaison filaire et sans fil. | 137 |
| Figure V.22 | Topologie réseau et fonctionnement du l'IEEE1588-PBS hybride..... | 138 |
| Figure V.23 | Le réseau simulé..... | 139 |
| Figure V.24 | La synchronisation entre C <-> Rs pour l'IEEE1588-PBS hybride..... | 140 |
| Figure V.25 | La synchronisation entre Rs <-> Ns pour l'IEEE1588-PBS hybride. | 140 |
| Figure V.26 | La consommation d'énergie pour les nœuds du réseau. | 141 |
| Figure V.27 | Synchronization du Niveau 1 et Niveau 2 avec Directionnal-TDMA MAC. | 143 |
| Figure V.28 | Consommation en énergie des routeurs..... | 144 |
| Figure V.29 | Consommation en énergie des nœuds 1588. | 144 |
| Figure V.30 | Consommation en énergie des nœuds PBS..... | 145 |
| Figure V.31 | Utilisation des antennes dans une séquence de synchronisation..... | 145 |

Liste des tableaux

| | | |
|----------------------|--|------------|
| Tableau I.1 | Caractéristiques pour des technologies de communications sans fil | 23 |
| Tableau I.2 | Comparaison des différents outils de simulation..... | 42 |
| Tableau I.3 | Les principaux composants du simulateur NS-2..... | 43 |
| Tableau II.1 | L'ensemble du travail à réaliser..... | 57 |
| Tableau II.4 | Description du cas d'utilisation relatif au nœud. | 63 |
| Tableau III.1 | Classification des protocoles de synchronisation. | 86 |
| Tableau IV.1 | Les paramètres généraux des antennes..... | 102 |
| Tableau IV.2 | Comparaison des protocoles réactifs et protocoles proactifs. | 111 |
| Tableau V.1 | Comparaison de valeurs obtenues pour la précision de la synchronisation. | 136 |
| Tableau V.2 | Les messages de synchronisation..... | 141 |