

Adaptation et mise en œuvre de la méthode d'accès MaCARI dans un réseau de capteurs sans fils : application à la surveillance dans une mine souterraine

Lionel Pascal

▶ **To cite this version:**

Lionel Pascal. Adaptation et mise en œuvre de la méthode d'accès MaCARI dans un réseau de capteurs sans fils : application à la surveillance dans une mine souterraine. Réseaux et télécommunications [cs.NI]. 2011. dumas-01101686

HAL Id: dumas-01101686

https://dumas.ccsd.cnrs.fr/dumas-01101686

Submitted on 9 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS
CENTRE REGIONAL DE CLERMONT-FERRAND

MÉMOIRE

présenté en vue d'obtenir

le DIPLÔME d'INGÉNIEUR CNAM
INFORMATIQUE
RÉSEAUX, SYSTÈMES ET MULTIMÉDIA
par

Lionel Pascal

Adaptation et mise en œuvre de la méthode d'accès MaCARI dans un réseau de capteurs sans fils.

Application à la surveillance dans une mine souterraine.

Soutenu le 15 juin 2011

Président :	K. BARKAOUI
Enseignant du Centre Associé :	M. MISSON
	M. SCHNEIDER
	A. FREITAS
	A. GUITTON
Ingénieurs :	P. LLAMAS
	P. LAFARGUETTE

Remerciements

En préambule de ce mémoire, je souhaite adresser mes remerciements à toutes les personnes qui ont contribué à faire de cette expérience une aventure unique. Mon stage s'étant déroulé au sein de deux laboratoires, les remerciements seront organisés en deux parties : l'une pour la période au LIMOS en France, l'autre pour la période au LRTCS au Canada.

Remerciements LIMOS

En premier lieu, je tiens sincèrement à remercier mon directeur de stage M. Michel Misson, Professeur au LIMOS, qui m'a accueilli dans son équipe et qui m'a apporté aide et soutien tout au long du stage. Je remercie également M. Gérard Chalhoub, mon encadrant technique, pour les indications qu'il m'a fourni.

Ensuite, je tiens à remercier l'ensemble des membres de l'équipe Réseaux et Protocoles pour leur accueil chaleureux. Je souhaite en particulier exprimer ma gratitute à M. Alexandre Guitton, Maître de Conférences, pour l'aide continue qu'il m'a apporté avec gentillesse et professionalisme, notamment lors de l'élaboration et de la relecture du mémoire. Egalement, je tiens à remercier M. Francois Delobel, Maître de Conférences, pour sa disponibilité et ses precieux conseils en matière de programmation système.

Puis, je souhaite remercier toutes les personnes et amis que j'ai cotoyé au laboratoire, en particulier, Nancy El Rachkidy, doctorante, pour sa bonne humeur et l'aide considérable qu'elle m'a fourni lors de la rédaction du mémoire. Je remercie également Hervé (mon camarade CNAM et ami de toujours), Nassima (pour sa maitrise de xfig et son humour kabyle), Chakib (Gibson forever !) et Stéphane.

Enfin, je remercie mes proches pour m'avoir apporté leur soutien durant cette période. Je pense particulièrement à ma mère, à qui je dédie l'ensemble de mon travail, et à toute ma famille, toujours unie dans les épreuves. J'embrasse, au passage, ma merveilleuse petite nièce Ariane, nouvellement entrée dans la famille.

Remerciements LRTCS

Je tiens tout d'abord à saluer l'excellente qualité de l'accueil qui m'a été fait au sein du LRCTS et souligner la bonne ambiance qui règne parmi ses occupants, faisant de ce laboratoire un lieu de travail agréable et enrichissant.

Je remercie M. Nadir Hakem, Professeur-chercheur à l'Université du Quebec de l'Abitibi-Témiscamingue pour sa disponibilité et ses conseils.

Je tiens également à remercier M. Mohammed Ailas, mon tuteur de projet, et Maxim Belleau, technicien au laboratoire qui m'a accompagné durant mes excursions minières. Merci également à M. Mourad Nedil et à ses précieuses explications sur les affres du comportement du signal et sur l'histoire de l'Algérie.

Enfin, je tiens à remercier particulièrement mon extraordinaire logeuse Linda, mes collocataires Carl, Jessica et Rosine ainsi que toute la "band" de fêtards (Jess, Carine, Kim...) avec qui j'ai passé des moments inoubliables.

Table des matières

1 Introduction **11**

1.1 Contexte 11

1.1.1 Le LIMOS 11

1.1.2 L'équipe Réseaux et Protocoles 12

1.2 Thématique abordée 13

1.2.1 Problématique 13

1.2.2 Le projet OCARI 15

1.2.3 Le protocole MaCARI 18

1.2.4 Travaux de l'équipe Réseaux et Protocoles sur MaCARI 19

1.3 Objectifs du stage 19

1.3.1 Problématique 20

1.3.2 Objectif 20

1.3.3 Déroulement 20

1.4 Organisation du mémoire 21

2 État de l'art **23**

2.1 Principaux standards pour les réseaux de capteurs sans fil 23

2.1.1 IEEE 802.15.4 24

2.1.1.1 Couche physique IEEE 802.15.4 . . . 25
2.1.1.2 Couche MAC IEEE 802.15.4 . . . 27
2.1.2 ZigBee . . . 32
2.1.3 Solutions alternatives . . . 37
2.1.3.1 WirelessHART . . . 37
2.1.3.2 ISA 100.11a . . . 39
2.1.3.3 Protocoles non standardisés . . . 41
2.1.4 Bilan . . . 41
2.2 MaCARI . . . 43
2.2.1 Synchronisation et découpage temporel . . . 45
2.2.2 Accès au médium . . . 48
2.2.3 Création du réseau et gestion de la topologie . . . 48
2.2.4 Affectation des adresses hiérarchiques . . . 49
2.2.5 Relais garanti pour une différenciation de service . . . 49
2.2.6 Estampillage temporel des trames . . . 50
2.2.7 Exemple de cycle global . . . 50
2.2.8 Acheminement de bout en bout en moins de quatre cycles . . . 50
2.2.9 Conclusion . . . 54
2.3 Plateforme de développement . . . 54
2.3.1 Plateforme matérielle Crossbow TelosB . . . 55
2.3.2 Le Système d'exploitation TinyOS et le langage NesC . . . 55

3 Contribution : implémentation de MaCARI sur TelosB 57
3.1 Base logicielle de l'implémentation de MaCARI . . . 57
3.1.1 Implémentation du standard IEEE 802.15.4 de l'Institut Polytechnique de Porto . . . 58
3.1.1.1 Fonctionnalités supportées par CISTER15.4 . . . 58
3.1.1.2 Architecture de CISTER15.4 . . . 58
3.1.1.3 Caractéristiques de CISTER15.4 . . . 60
3.1.2 Implémentation du standard IEEE 802.15.4 de l'Université Technique de Berlin . . . 60

3.1.2.1 Fonctionnalités supportées par TKN15.4 61
3.1.2.2 Architecture de TKN15.4 61
3.1.2.3 Caractéristiques de TKN15.4 64
3.1.3 Bilan 65
3.2 Implémentation de MaCARI sur TelosB 66
3.2.1 Fonctionnalités 66
3.2.2 Architecture et mécanismes de l'implémentation 67
3.2.2.1 Les composants et les interfaces de la couche PHY . . . 69
3.2.2.2 Les composants et les interfaces de la couche MaCARI . 69
3.3 Adaptation et optimisation de l'implémentation de MaCARI 78
3.3.1 Phase de chargement de trame dans le *buffer* d'envoi du module radio 78
3.3.2 Période de synchronisation 82

4 Évaluation **85**
4.1 Synchronisation 85
4.2 Les méthodes d'accès 89
4.2.1 Phase de transmission physique des trames 89
4.2.2 Envoi direct 91
4.2.3 Envoi avec CSMA/CA slotté 91
4.2.3.1 Débit maximum 91
4.2.3.2 Période de collecte 92
4.2.3.3 Période de routage 94
4.3 Délai de bout en bout 96
4.3.1 Durée d'un cycle MaCARI 97
4.3.2 Trafic prioritaire à destination du CPAN 99
4.3.3 Trafic prioritaire à destination quelconque 99
4.4 Consommation énergétique 100

5 Mise en œuvre **105**
5.1 Contexte 105
5.1.1 Le laboratoire de recherche Télébec en communication souterraines 106

5.1.2 La mine-laboratoire CANMET 106
5.1.3 Les mines souterraines et l'industrie minière 107
5.2 Description de l'application et de son implémen-tation 108
5.2.1 Description de l'application 108
5.2.2 Implémentation de l'application au dessus du protocole MaCARI 109
5.3 Configuration de MaCARI 111
5.4 Étude de portée et déploiement dans la mine 113
5.4.1 Étude de la portée des cartes TelosB 113
5.4.1.1 Portée en milieu ouvert sans obstacle 113
5.4.1.2 Portée en milieu souterrain 114
5.4.1.3 Conclusion sur la portée des cartes TelosB en milieu souterrain 118
5.4.2 Déploiements dans la mine 118
5.5 Évaluation de l'application 120
5.5.1 Étude du trafic non prioritaire 121
5.5.2 Évaluation du délai de bout en bout du mécanisme d'alarme . . . 123
5.5.3 Consommation énergétique 125
5.5.3.1 Bilan 126

6 Conclusion **127**
6.1 Contributions 128
6.2 Perspectives 129

A Correspondance technique **131**

B Liste des abreviations **143**

C Glossaire **147**

Table des figures

1.1 Exemple d'un réseau de capteurs sans fil relié à un système de contrôle *via* une passerelle. 14
1.2 Topologie du réseau industriel OCARI. 16
1.3 Topologie du réseau industriel OCARI. 17
1.4 Le cycle de MaCARI est organisé en trois périodes. 18

2.1 Types de topologies disponibles pour le standard IEEE 802.15.4. 25
2.2 Représentation des canaux disponibles pour le standard IEEE 802.15.4. . 26
2.3 Représentation des différentes périodes d'un cycle global. 28
2.4 Algorithme du CSMA/CA slotté. 29
2.5 Algorithme du CSMA/CA non slotté. 31
2.6 Représentation de la pile IEEE 802.15.4/ZigBee. 32
2.7 Réseau avec une topologie arborescente. 34
2.8 Exemple de topologie arborescente et de l'adressage hiérarchique. 35
2.9 Exemple d'un réseau WirelessHART maillé. 38
2.10 Exemple d'un réseau ISA100 connecté au réseau d'une usine. 40
2.11 Exemple de topologie d'un réseau OCARI. 44
2.12 Le cycle global de MaCARI. 45

2.13 Principe de la cascade de *beacon* dans MaCARI. 46
2.14 Période d'activité suivie par un intervalle de relais. 47
2.15 Décomposition d'un cycle MaCARI. 51
2.16 Représentation du délai d'acheminement maximum de bout en bout dans MaCARI. 53
2.17 Carte Crosbow TelosB. 55

3.1 Organisation de l'architecture de l'implémentation CISTER15.4. 59
3.2 Représentation des évènements du *timer*. 59
3.3 Représentation des composants logiciels de la couche MAC TKN15.4. . . 62
3.4 Représentation d'un exemple de transmission du jeton entre les composants. 64
3.5 Représentation de l'architecture de l'implémentation de MaCARI. 68
3.6 Représentation des mécanismes de passage de jeton et de l'accès à la ressource radio. 70
3.7 Représentation du rôle des composants durant la période de synchronisation. 72
3.8 Représentation de la gestion du jeton par l'ordonnanceur des périodes d'activité. 74
3.9 Représentation des composants responsables des échanges de données durant l'intervalle $[T_1; T_3]$. 75
3.10 Représentation de la répartition des trames dans les files d'attente. . . . 77
3.11 Représentation du bus SPI reliant le module radio et le microprocesseur. 79
3.12 Temps de chargement d'une trame dans le *buffer* d'envoi. 79
3.13 Impact du DMA sur le temps de chargement dans le *buffer* d'envoi. . . . 81
3.14 Impact du cadencement du DCO à 8 MHz sur le temps de chargement. . 81
3.15 Les trois phases qui composent un intervalle de cascade de beacons. . . . 83
3.16 Évaluation de la durée cumulée des trois phases délimitant un intervalle de cascade de *beacons*. 83

4.1 Représentation de la topologie arborescente utilisée pour évaluer la synchronisation. 86
4.2 Exemple de mesures réalisées avec l'analyseur d'états logiques sur la précision de la synchronisation dans un réseau MaCARI. 87

4.3 Évolution de la synchronisation par rapport au CPAN en fonction de la durée du cycle, pour la topologie en arbre illustrée sur la figure 4.1. . . . 88
4.4 Durée de transmission effective en fonction de la taille des trames envoyées. 90
4.5 Débit en fonction de la taille des trames envoyées. 90
4.6 Évaluation du débit maximal selon la méthode d'accès. 92
4.7 Photographie de cartes TelosB formant une topologie en étoile. 93
4.8 Nombre de trames reçues par le coordinateur en fonction du nombre de feuilles et du nombre de trames émises par chaque feuille (trames de 50 octets avec acquittement). 93
4.9 Topologie linéaire simple pour évaluer la période $[T_2; T_3]$. 94

5.1 Coupe transversale de la mine-laboratoire CANMET. 107
5.2 Galerie d'une mine d'or à Val d'or (Canada). 108
5.3 Topologie du réseau linéaire utilisé pour l'expérimentation de l'application. 111
5.4 Étude de la portée en milieu ouvert sans obstacle. 114
5.5 Disposition dans la mine de l'expérimentation, en LOS. 115
5.6 Mesures du RSSI, du taux de trames reçues et du LQI, en LOS. 115
5.7 Étude de la portée dans un virage de la rampe, en NLOS. 116
5.8 Mesures du RSSI, du taux de trames reçues et du LQI, en NLOS. 116
5.9 Étude de la portée dans le cas d'un relief inégal, en NLOS. 117
5.10 Mesures du RSSI, du taux de trames reçues et du LQI, pour un relief inégal, en NLOS. 117
5.11 Boitier étanche contenant une carte TelosB (en haut à gauche), et photos de la disposition du CPAN dans la mine. 118
5.12 Disposition des nœuds pour le déploiement A dans la mine. 119
5.13 Disposition des nœuds pour le déploiement B dans la mine. 120
5.14 Étude du trafic non prioritaire généré par le coordinateur C_{10} durant $[T_2; T_3]$. 122
5.15 Étude du trafic non prioritaire généré par trois coordinateurs géographiquement éloignés durant $[T_2; T_3]$. 122
5.16 Représentation des délais d'acheminement mesurés entre l'émission et la réception de chaque alarme émise. 124

Liste des tableaux

1.1 Effectif du LIMOS. 12
1.2 Classification des applications selon l'importance opérationnelle des messages. 15

2.1 Débit selon la modulation et la fréquence. 26
2.2 Valeur de $Cskip$ selon la profondeur du nœud pour $L_m = 3$, $C_m = 3$ et $R_m = 2$. 35
2.3 Comparatif entre les standards. 42

3.1 Description des fonctions fournies par les composants logiciels de TKN15.4. 63
3.2 Bilan comparatif entre CISTER15.4 et TKN15.4. 66

4.1 État du module radio au cours d'un cycle MaCARI selon la période. . . . 101
4.2 Consommation énergétique d'une carte TelosB selon l'état du module radio. 102

5.1 Résultats de l'expérimentation du mécanisme d'alarme. 124

Chapitre 1

Introduction

Ce mémoire constitue la synthèse des travaux réalisés pendant le stage de dix mois que j'ai effectué dans le cadre de la formation d'ingénieur CNAM en informatique, au sein d'un laboratoire de recherche en informatique. Ce chapitre décrit tout d'abord le laboratoire et ses activités, puis il introduit le projet de recherche abordé et sa problématique. Ensuite il présente l'objectif du stage et mon rôle dans l'équipe. Enfin, ce chapitre détaille l'organisation du mémoire.

1.1 Contexte

Ce stage s'est déroulé essentiellement au sein de l'équipe Réseaux et Protocoles du LIMOS.

1.1.1 Le LIMOS

Le LIMOS (Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes) est une Unité Mixte de Recherche (UMR) CNRS, dépendant du Département ST2I (Sciences et Technologies de l'Information et de l'Ingénierie). Situé à Clermont-Ferrand, c'est un laboratoire interuniversitaire rattaché à l'Université d'Auvergne, l'Université

Blaise Pascal et à l'IFMA (Institut Français de Mécanique Avancée).

Le LIMOS aborde des thématiques qui reposent sur l'étude et la recherche de modèles et d'outils informatiques destinés à des systèmes complexes (systèmes de transport, de télécommunications, de production, écosystèmes...). Son activité est divisée en trois axes de recherche. Le premier axe est lié à la recherche opérationnelle et à l'aide à la décision, le second aux systèmes d'information et de communication et le troisième à la modélisation, à l'organisation et au pilotage des systèmes de production. Le personnel du laboratoire est organisé en plusieurs équipes totalisant environ 140 personnes se répartissant comme indiqué dans le tableau 1.1.

Tableau 1.1 – Effectif du LIMOS.

Professeurs	13
Maîtres de Conférences	51
MCF et Professeurs habilités	13
Ingénieur	1
Techniciens	2
Secrétaire	1
Doctorants	54
Post-doctorants	5

1.1.2 L'équipe Réseaux et Protocoles

Le deuxième axe du LIMOS, lié aux systèmes d'information et de communication, intègre l'équipe de recherche en informatique Réseaux et Protocoles. Elle est dirigée par M. Misson, professeur à l'Université d'Auvergne, et compte actuellement cinq maîtres de conférences et quatre doctorants.

Les cinq thèmes abordés dans l'équipe s'inscrivent dans le domaine des réseaux informatiques basés sur des technologies sans fil, traitant plus spécifiquement les couches basses (couches liaison de données, MAC et couche réseau) ainsi que les applications de localisation en milieu confiné. La liste des thèmes abordés par l'équipe est la suivante :

- Thème 1 : Étude de l'intégration de solutions 802.11 en milieu industriel.
- Thème 2 : Déploiement d'un réseau de capteurs étendu et linéaire.
- Thème 3 : Modèle de réseaux de type LP-WPAN pour des applications de traçabilité des contacts d'une population mobile.

– Thème 4 : Mise en œuvre d'un protocole réseau déterministe et économe en énergie pour la surveillance d'un site industriel.
– Thème 5 : Conception de protocoles MAC réactifs et très économes en énergie.

1.2 Thématique abordée

Nous allons à présent détailler la problématique du quatrième thème de l'équipe, à savoir la mise en œuvre d'un protocole réseau déterministe et économe en énergie pour la surveillance d'un site industriel. Nous situerons ensuite cette thématique dans le cadre du projet de recherche OCARI (pour Optimisation des Communications Ad-hoc pour les Réseaux Industriels). Ensuite, nous décrirons le protocole de gestion d'accès au médium MaCARI. Enfin nous aborderons les objectifs et les réalisations de l'équipe Réseaux et Protocoles concernant MaCARI qui est au cœur du travail que j'ai mené.

1.2.1 Problématique

Certains bâtiments et édifices industriels nécessitent que soient surveillés leur évolution, dégradation ou vieillissement. Pour cela, des mesures doivent être effectuées afin d'évaluer les conséquences des évolutions de ces installations et vérifier si les conditions de sécurité sont toujours respectées. Ces données peuvent être récoltées par un réseau de capteurs et acheminées à un système de contrôle. Or, les installations industrielles ne peuvent généralement pas accueillir facilement la pose de câbles qui relient et alimentent les capteurs (cette opération peut être très couteuse : difficulté d'accès, problèmes de sécurité...). Par exemple, les centrales nucléaires, les barrages hydrauliques ou les mines souterraines peuvent difficilement supporter un réseau filaire supplémentaire. La solution consiste donc à utiliser un réseau de capteurs munis de module radio permettant une communication sans fil, ce qui simplifie l'installation du réseau.

Un réseau de capteurs sans fil est un ensemble d'entités communicantes (appelées nœuds) dont les principales propriétés sont : une capacité de calcul généralement assurée par un microcontrôleur, une consommation énergétique réduite, une capacité de communication radio, un faible coût de fabrication, une taille réduite, un ou des capteurs et/ou actionneurs (de température ou de luminosité par exemple). Ces entités commu-

niquent entre elles *via* la modulation d'une onde radio pour échanger des informations. La figure 1.1 présente un exemple de topologie d'un réseau de capteurs dans lequel des informations peuvent être acheminées de proche en proche jusqu'à un système de contrôle accessible en filaire par le biais d'une passerelle (entité capable d'interconnecter des réseaux distincts).

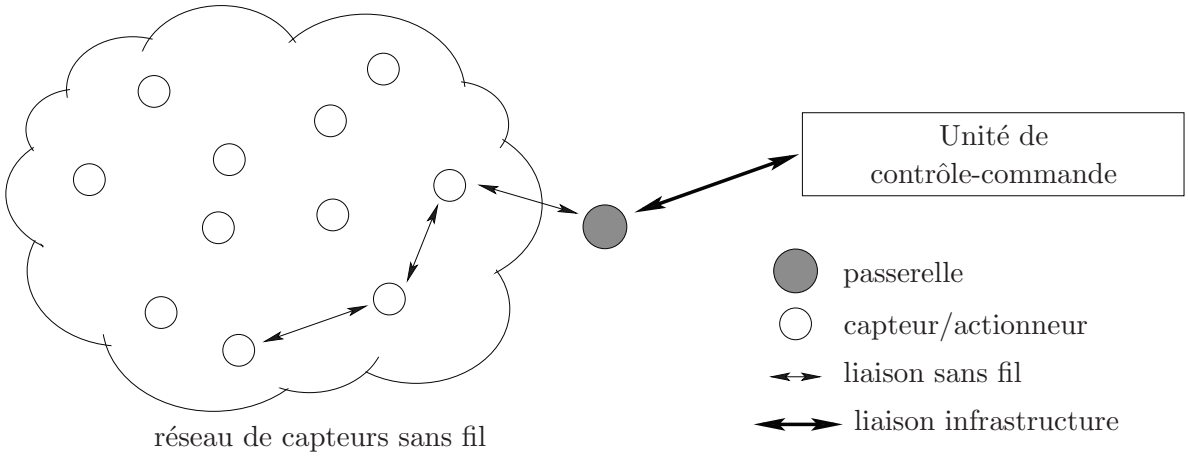


Figure 1.1 – Exemple d'un réseau de capteurs sans fil relié à un système de contrôle *via* une passerelle.

Les réseaux de capteurs bénéficient des récents progrès technologiques au niveau de la miniaturisation et d'un panel de différents types de capteurs. La variété des possibilités qu'ils offrent est étendue : traçabilité, surveillance, transmission de données environnementales, mobilité, modularité, facilité d'installation, etc [1].

Cependant l'architecture des installations industrielles et la présence de structures métalliques perturbent souvent la propagation des ondes radio, ce qui a un impact fort sur le taux de pertes d'information et sur le délai d'acheminement quand les informations doivent être répétées. Cela soulève un problème pour les informations critiques comme les alarmes pour lesquelles le délai et les pertes ont une incidence sur le fonctionnement et la sécurité des installations. Une classification des applications industrielles a été réalisée [2]. Elle est organisée en fonction de l'importance de la réactivité et de la fiabilité de l'acheminement des informations.

Le tableau 1.2 décrit des catégories d'applications associées à des classes de criticité afin de situer le champ d'application d'un réseau selon ses caractéristiques. Par nature, les réseaux de capteurs sans fil sont destinés à des applications de classe 3, 4 et 5,

Tableau 1.2 – Classification des applications selon l'importance opérationnelle des messages.

Catégorie	**Classe**	**Application**	**Importance du délai d'acheminement des messages pour que l'application reste opérationnelle**
Sécurité	0	Commande d'urgence	Messages toujours critiques
Contrôle	1	Contrôle régulier nécessitant une réactivité élevée	Messages souvent critiques
	2	Contrôle régulier et supervision	Messages parfois critiques
	3	Contrôle irrégulier	Messages ponctuels rarement critiques
Surveillance	4	Alerte	Messages tolérant un court délai
	5	Transit de données	Pas de conséquence opérationnelle en cas de délai important

non critiques, et dont le débit est limité. Ces classes correspondent aux besoins de surveillance d'installations industrielles et éventuellement à des applications où le contrôle est irrégulier (par exemple, la commande à distance et le changement d'état d'un actionneur ou la relève d'une mesure sur un type de capteur donné). Pour toutes ces applications, afin d'optimiser les performances du réseau, il est important que le déploiement soit le sujet d'une étude précise afin de placer judicieusement les capteurs pour ne pas gêner le fonctionnement de l'installation, mais aussi pour limiter le nombre de sauts que doit effectuer l'information avant d'atteindre sa destination. L'autonomie des entités est aussi une contrainte forte : l'autonomie du réseau doit être maximale afin de limiter la maintenance (changement des piles) à des périodes réalistes en termes de coût et de temps.

Les problématiques évoquées de topologie du réseau, de limitation du délai de l'acheminement de l'information et d'autonomie pour des installations industrielles sont fortement liées au protocole de communication utilisé, c'est-à-dire aux mécanismes utilisés par les entités du réseau pour échanger des informations.

1.2.2 Le projet OCARI

La thématique de mon stage est associée au projet OCARI [3] (Optimisation des Communications Ad-hoc pour les Réseaux Industriels) basé sur un partenariat entre

académiques et industriels. Il a pour but de proposer une nouvelle génération de réseau sans fil permettant la surveillance d'installations et d'édifices industriels. L'objectif du projet OCARI est de définir une pile réseau complète dans laquelle chaque couche contribue à fournir les éléments capables de répondre aux besoins identifiés par les industriels [4].

Un réseau OCARI est composé d'îlots séparés physiquement (soit géographiquement, soit en utilisant des canaux différents) et pouvant chacun contenir jusqu'à 400 nœuds. Chacun de ces îlots possède une structure arborescente et une passerelle permettant son interconnexion éventuelle à une infrastructure (filaire ou non). Un nœud mobile (appelé rondier) peut être également utilisé pour véhiculer des informations entre les îlots. La figure 1.2 présente trois ilots interconnectés par le biais de passerelles reliées à une unité de contrôle. Le rondier peut se connecter de manière ponctuelle à l'un des ilots afin d'effectuer des interventions localisées ou de collecter des informations.

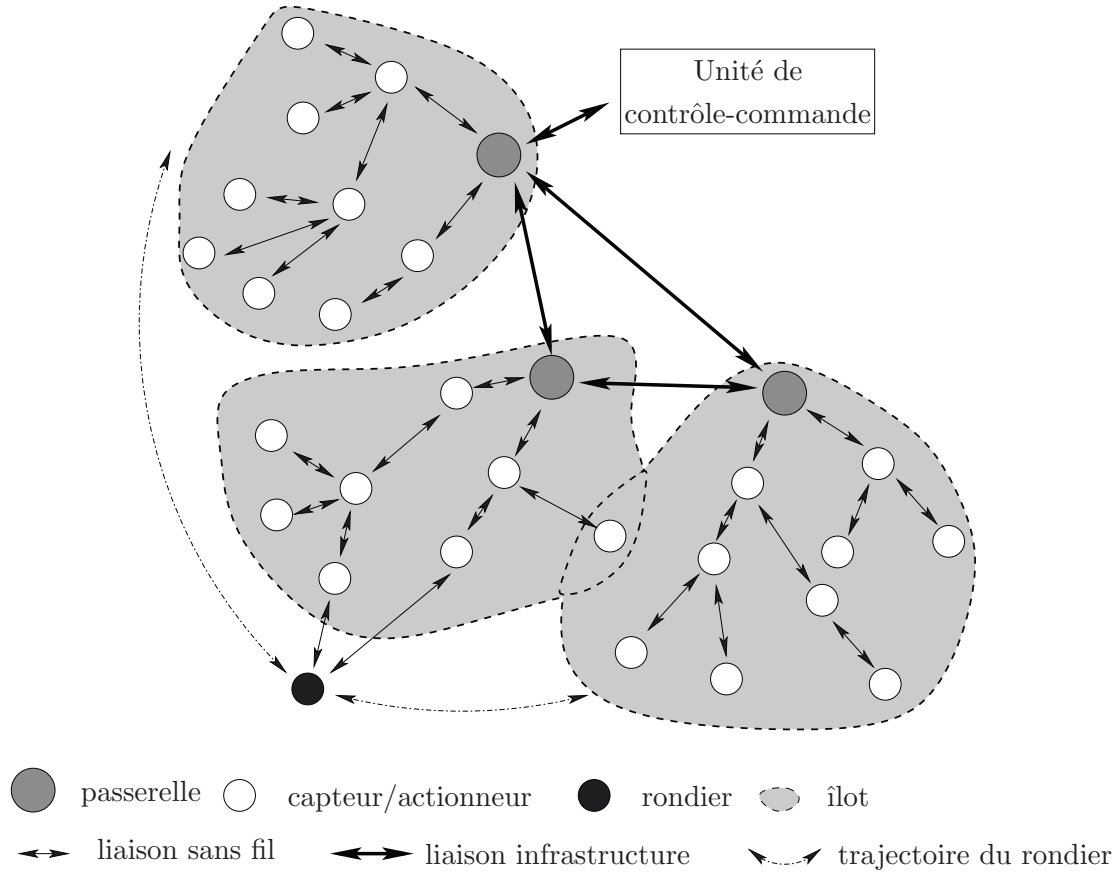


Figure 1.2 – Topologie du réseau industriel OCARI.

Le projet OCARI est subventionné par l'ANR [5] (Agence Nationale de la Recherche) et regroupe quatre partenaires académiques, à savoir l'INRIA (Institut National de Re-

cherche en Informatique et Automatique), le LATTIS (Laboratoire Toulousain de Technologie et d'Ingénierie des Systèmes), le LIMOS et le LRI (Laboratoire de Recherche en Informatique). Le projet compte également trois partenaires industriels. Le premier, ÉDF R&D (Électricité de France, Recherche et Développement) compte équiper de réseaux de capteurs ses sites de production d'énergie tels que les barrages hydrauliques et les centrales nucléaires. La DCNS (Direction de Construction Navale Sextant) a pour volonté de développer une technologie réseau à bord de navires de guerre constituant un environnement fortement contraint. Enfin, Télit propose des prototypes de cartes et un savoir-faire au niveau de leur programmation.

La figure 1.3 montre la répartition des rôles entre les différents acteurs du projet selon leur contribution sur la pile OCARI. Le travail de l'équipe Réseaux et Protocoles du LIMOS est principalement axé sur l'élaboration d'un protocole d'accès au médium nommé MaCARI, et sur la réalisation du maquettage de ce protocole. Ce travail doit être réalisé en conservant cohérence et compatibilité vis-à-vis des protocoles et mécanismes produits par les autres partenaires.

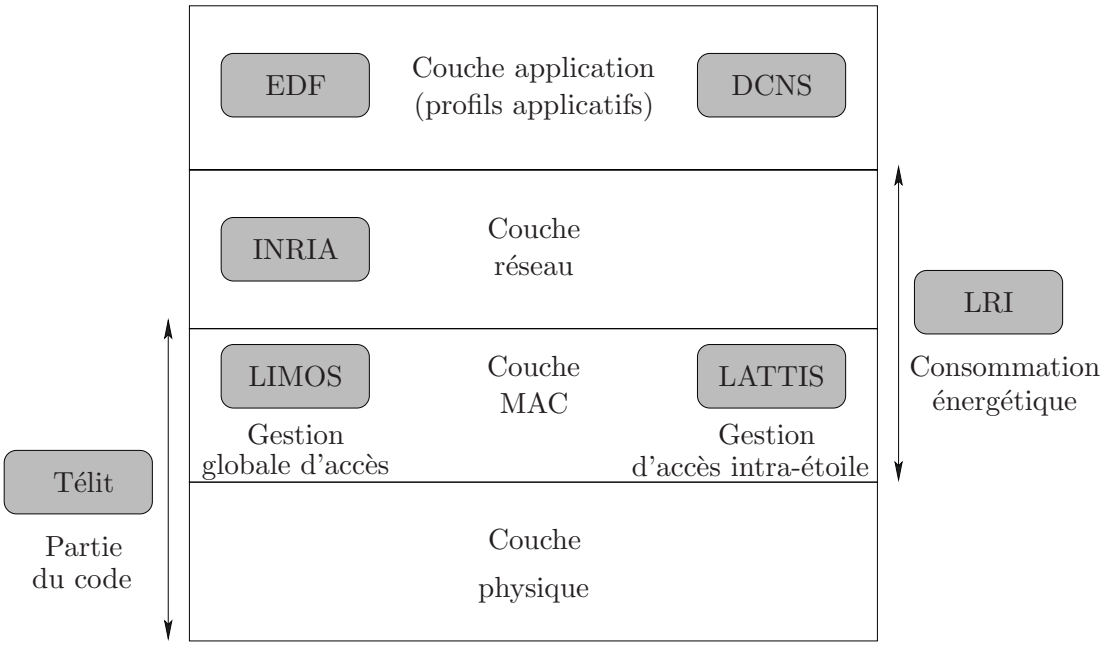


Figure 1.3 – Topologie du réseau industriel OCARI.

1.2.3 Le protocole MaCARI

MaCARI (MAC pour OCARI) est un protocole d'accès au médium destiné aux réseaux de capteurs sans fil créé principalement par l'équipe Réseaux et Protocoles du LIMOS. Ce protocole, qui prend en compte les limitations des entités du réseau, est à la fois déterministe et économe en énergie. Le déterminisme permet de garantir l'accès au médium. L'économie d'énergie consiste à optimiser l'activité de chaque entité du réseau pour faire en sorte qu'elle soit en état de sommeil le plus souvent possible, afin de prolonger au maximum l'autonomie de l'entité et par conséquent celle du réseau.

MaCARI propose une différenciation de trafic et un délai borné d'acheminement des messages : le trafic constitué des informations associées à un délai de transit borné et le trafic des informations dont le délai de transit est non borné dans le temps. Une métaphore souvent utilisée explique les principes du protocole : on peut assimiler les communications dans MaCARI à la remontée des rivières par les saumons. Lorsqu'ils remontent la rivière, les saumons sont parfois confrontés à des obstacles comme des cascades. Si les saumons ne parviennent pas à franchir un obstacle trop élevé, on peut mettre en place des « échelles à saumons », qui permettent aux saumons de poursuivre avec certitude leur remontée au prix d'un délai plus élevé. Ainsi, les saumons ont deux options : essayer de remonter la cascade ou emprunter l'échelle. Dans MaCARI, une information peut être envoyée de deux manières : soit par une méthode qui ne garantit pas de délai jusqu'à sa destination, soit par une méthode qui garantit que la destination recevra le message dans un délai borné.

MaCARI est organisé en cycle dans lequel alternent la période de synchronisation, la période d'activité et la période d'inactivité. Ce découpage permet notamment d'organiser la gestion de la mise en sommeil des entités du réseau. La figure 1.4 donne une illustration d'un cycle MaCARI.

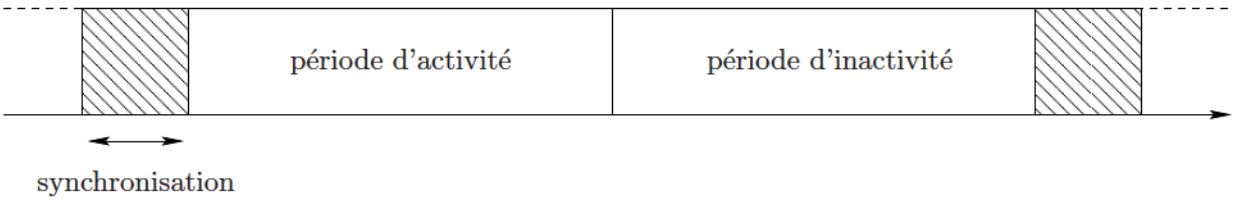


Figure 1.4 – Le cycle de MaCARI est organisé en trois périodes.

L'élaboration du protocole MaCARI est issue d'une réflexion approfondie sur l'étude des protocoles MAC existants. MaCARI repose sur la couche physique du standard IEEE 802.15.4 destinée aux réseaux de capteurs sans fil et propose de nombreuses modifications de ce standard IEEE 802.15.4, principalement au niveau de la sous-couche MAC et de la synchronisation des nœuds. Les descriptions du standard IEEE 802.15.4 et du protocole MaCARI seront abordées dans le deuxième chapitre du mémoire.

1.2.4 Travaux de l'équipe Réseaux et Protocoles sur MaCARI

L'élaboration de MaCARI a été coordonnée avec les différents partenaires. L'équipe Réseaux et Protocoles a mené une réflexion sur les caractéristiques du protocole. Ensuite, MaCARI a été spécifié puis porté sur Network Simulator 2 [6], un logiciel permettant de simuler le comportement d'un protocole selon différents scénarios. De plus, un outil graphique (nommé MaCAVIEW) permettant de mieux visualiser les simulations a été développé. Enfin, MaCARI a été porté sur des prototypes de cartes B2400ZB-tiny [7] provenant du partenaire Télit, ceci en modifiant en profondeur l'implémentation du standard IEEE 802.15.4 déjà présente sur les cartes. Cette maquette a été présentée aux partenaires industriels avec succès en juin 2009. Le détail et les résultats de ces travaux ont été transcris de manière structurée dans la thèse de G. Chalhoub [8] préparée dans cette équipe et soutenue en décembre 2009.

Depuis, MaCARI fait l'objet de plusieurs sous-projets de recherche et de développement supervisés par l'équipe Réseaux et Protocoles. C'est dans ce cadre que des ingénieurs, des stagiaires ingénieurs CNAM et des élèves de l'université sont amenés à intégrer ponctuellement l'équipe.

1.3 Objectifs du stage

Cette partie présente la problématique puis les objectifs de mon stage, mon rôle et enfin les différentes étapes de son déroulement.

1.3.1 Problématique

Une version de MaCARI existait sur les cartes du partenaire Télit avant le début de mon stage (celle ayant servie pour la maquette du réseau). Le langage et les outils de développement utilisés pour programmer les cartes sont imposés par le constructeur. Cette dépendance peut s'avérer être une contrainte forte pour l'ensemble du projet OCARI car le système et le langage du code utilisé pour cette implémentation sont propriétaires. L'outil de développement est protégé : il est nécessaire de payer une licence pour pouvoir programmer sur les cartes. D'autre part, la propriété intellectuelle du code appartient en partie à Télit. L'équipe s'est donc orientée sur l'étude d'une solution basée sur un type de carte Crossbow TelosB qui fournit une plateforme complètement *open-source* (libre accès au code source et libre droit à la redistribution) dont les caractéristiques sont décrites en 2.3.

1.3.2 Objectif

L'objectif de ce stage est de réaliser une adaptation du protocole MaCARI, de la porter sur les prototypes de cartes TelosB, d'expérimenter cette implémentation et enfin de mettre en œuvre son application en milieu industriel confiné.

Ce travail a été supervisé par mon tuteur de stage M. Misson et orienté par G. Chalhoub, mon encadrant technique. De plus, de nombreuses réunions avec l'ensemble de d'équipe m'ont permis d'avancer sur les points critiques de mon projet.

Mon rôle dans l'accomplissement des objectifs de ce stage est donc articulé autour des étapes d'étude, de développement, d'expérimentation et de mise en œuvre propres au métier d'ingénieur.

1.3.3 Déroulement

Le déroulement de mon stage, d'une durée de dix mois (soit d'octobre 2009 à juillet 2010), peut se découper en quatre phases : une phase de découverte du sujet et du domaine, une phase de réflexion sur l'adaptation de MaCARI, une phase de réalisation et d'expérimentation d'une implémentation de MaCARI, et enfin une phase de mise en œuvre dans une mine expérimentale au Canada.

La première phase était consacrée à la découverte et à l'appropriation de la problématique de ce sujet et de son environnement. À l'aide de la bibliographie fournie par M. Misson, j'ai pu aborder le domaine des réseaux de capteurs sans fil. J'ai étudié MaCARI à travers la thèse de G. Chalhoub qui m'a lui-même guidé à de nombreuses reprises.

Durant la phase d'adaptation, j'ai étudié deux versions différentes de la norme IEEE 802.15.4 disponibles pour les plateformes TelosB afin d'en choisir une comme base pour MaCARI. À cette occasion, je me suis familiarisé avec la programmation en langage NesC *(Network embedded system C)* utilisé par le système d'exploitation TinyOS *(Tiny Operating System)* présent sur les cartes.

Lors de la phase de réalisation et d'expérimentation, j'ai réalisé une implémentation de MaCARI fournissant les principales fonctionnalités utiles à mes objectifs. J'ai ensuite procédé à une série d'expérimentations afin d'étudier le comportement de mon implémentation pour vérifier s'il était cohérent avec les simulations et l'implémentation sur les cartes Télit.

Enfin, pour la phase de mise en ouvre, je me suis rendu à Val-d'Or au Canada, dans le cadre d'un stage de deux mois initié par M. Misson, au sein du LRTCS (Laboratoire de Recherche Télébec en Communications Souterraines). J'ai pu mettre en œuvre la maquette d'une application de surveillance de la ventilation dans une mine expérimentale.

1.4 Organisation du mémoire

La suite de ce mémoire est composée de cinq parties. L'état de l'art présente le standard IEEE 802.15.4 et ses alternatives, le protocole MaCARI et la plateforme matérielle TelosB. Ensuite, la partie contribution détaille le choix de la base utilisée pour l'implémentation, sa structure ainsi que les optimisations réalisées. Puis, la partie évaluation décrit selon plusieurs critères le comportement de l'implémentation. Suit la mise en œuvre d'une application de cette implémentation dans une mine canadienne. Enfin, la conclusion résume les contributions apportées avant de proposer différentes perspectives aux travaux réalisés.

Chapitre 2

État de l'art

L'objectif de ce chapitre est de présenter toutes les notions nécessaires à la compréhension de l'implémentation de MaCARI sur les cartes Crossbow TelosB.

Dans la première partie de ce chapitre, nous décrivons les principaux standards existants pour les réseaux de capteurs sans fil. Puis, nous présentons dans la deuxième partie, le protocole MaCARI et ses principes. Enfin, la troisième partie propose une description de la plateforme matérielle TelosB sur laquelle sera portée l'implémentation de MaCARI.

2.1 Principaux standards pour les réseaux de capteurs sans fil

Un standard pour les réseaux de capteurs sans fil est un ensemble de protocoles de communication normalisés, organisé en couches imbriquées et constituant une pile protocolaire. Les couches hautes de cette pile ont des fonctions liées à l'application tandis que les couches basses sont responsables de l'acheminement des données et doivent introduire des mécanismes adaptés aux communications sans fil. Pour cette étude, nous nous intéresserons à la couche physique (PHY) qui assure la transmission effective des données, à la couche de liaison de données qui organise la gestion de l'accès au médium

(MAC) et à la couche réseau qui assure le routage de l'information.

Il existe plusieurs standards destinés aux réseaux de capteurs sans fil industriels assurant l'échange d'informations entre les différentes entités. Ils prennent en compte les contraintes liées à l'autonomie, à la faible puissance de calcul et à l'espace mémoire limité des entités.

La majorité des protocoles pour les réseaux de capteurs sans fil dédiés au milieu industriel est basée sur la couche physique du standard IEEE 802.15.4. Tout d'abord, nous décrivons le standard IEEE 802.15.4 et le standard ZigBee qui ont tous deux servis de base à MaCARI. Ensuite, nous présentons les solutions alternatives majeures. Enfin, nous résumons par le biais d'un bilan comparatif les points forts et points faibles de l'ensemble des standards abordés.

2.1.1 IEEE 802.15.4

Le standard le plus répandu en milieu industriel pour les réseaux de capteurs sans fil est le standard IEEE 802.15.4 [9] qui définit les couches basses (PHY, LLC et MAC) de la pile protocolaire. Cette norme, adaptée à des équipements à faible consommation et à capacité réduite, convient à des applications nécessitant un débit peu élevé.

Le standard IEEE 802.15.4 supporte deux types de topologies : les topologies en étoile et les topologies maillées. Une topologie en étoile est composée d'un nœud central, dit coordinateur du PAN *(Personal Area Network)*, et de nœuds feuilles. Chaque feuille peut uniquement communiquer avec le coordinateur du PAN. Le coordinateur du PAN gère la connexion (association) et la déconnexion (désassociation) des feuilles ; la taille du réseau est donc limitée à sa portée. Une topologie maillée est composée de trois types de nœuds : le coordinateur du PAN, des coordinateurs et des feuilles. Chaque entité peut communiquer avec les nœuds qui sont à sa portée. Le coordinateur du PAN a un ou plusieurs coordinateurs fils qui peuvent eux aussi gérer l'association et la désassociation de feuilles et de coordinateurs. Le réseau est alors plus complexe et sa portée plus étendue. Ces deux topologies sont illustrées dans la figure 2.1.

D'un point de vue matériel, les réseaux sont composés de deux types d'équipements : les FFD *(Full Function Device)*, qui disposent de toutes les fonctionnalités disponibles

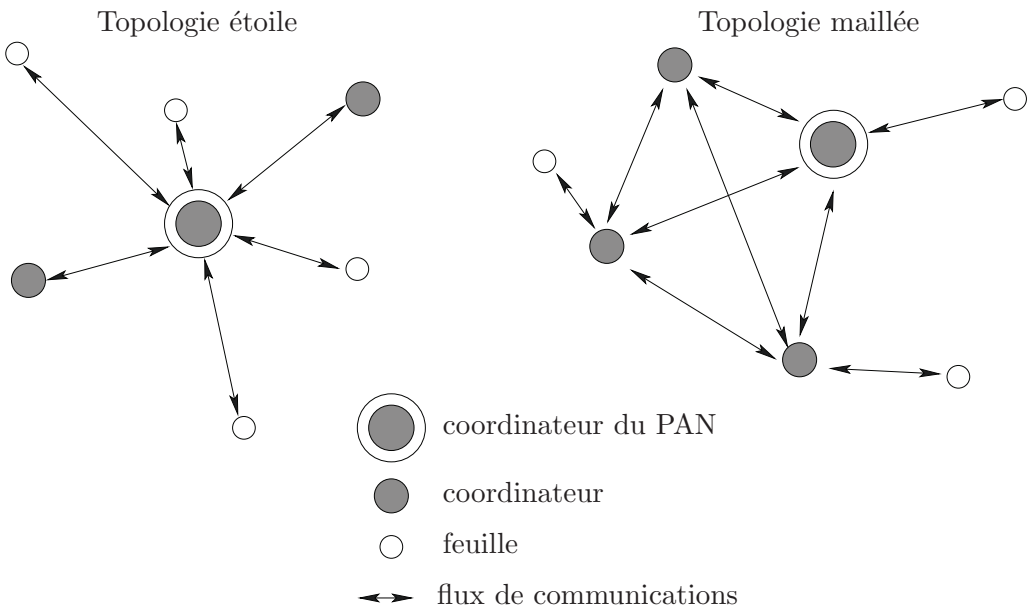


Figure 2.1 – Types de topologies disponibles pour le standard IEEE 802.15.4.

dans la norme, et les RFD *(Reduced Function Device)*, qui sont des entités allégées en fonctionnalités, donc moins coûteuse, et dont le rôle applicatif est simple. Ainsi, un FFD peut occuper, selon les besoins, le rôle de coordinateur du PAN, de coordinateur ou de feuille. Un RFD, en revanche, ne peut qu'avoir le rôle d'une feuille (sans fonctionnalité de routage).

2.1.1.1 Couche physique IEEE 802.15.4

La couche physique IEEE 802.15.4 fournit les services de transmission de données aux couches supérieures. Elle est généralement prise en charge par le module radio et réalise la sélection des canaux, la modulation du signal et gère la puissance d'émission. Les informations (découpées en trames de tailles variables par la couche MAC) sont envoyées selon une modulation donnée. Le débit proposé varie en fonction de la fréquence et de la modulation [10] comme le montre le tableau 2.1. L'utilisation des fréquences dépend de l'environnement et de la législation (variable selon les pays).

Les fréquences basses permettent généralement d'atteindre de plus grandes portées (due à de plus faibles pertes lors de la propagation). Les fréquences plus grandes permettent d'obtenir une latence plus faible, un nombre de canaux plus important (à bande de fréquences égale), mais pour une portée réduite.

Tableau 2.1 – Débit selon la modulation et la fréquence.

Fréquence (MHz)	**Modulation**	**Débit théorique maximum (kbps)**
$868 - 868.6$	BPSK	20
$868 - 868.6$	ASK	250
$868 - 868.6$	O-QPSK	100
$902 - 928$	BPSK	40
$902 - 928$	ASK	250
$902 - 928$	O-QPSK	250
$2400 - 2483.5$	O-QPSK	250

La couche physique IEEE 802.15.4 offre au total 27 canaux sur l'ensemble des trois bandes de fréquences. La figure 2.2 décrit leur répartition.

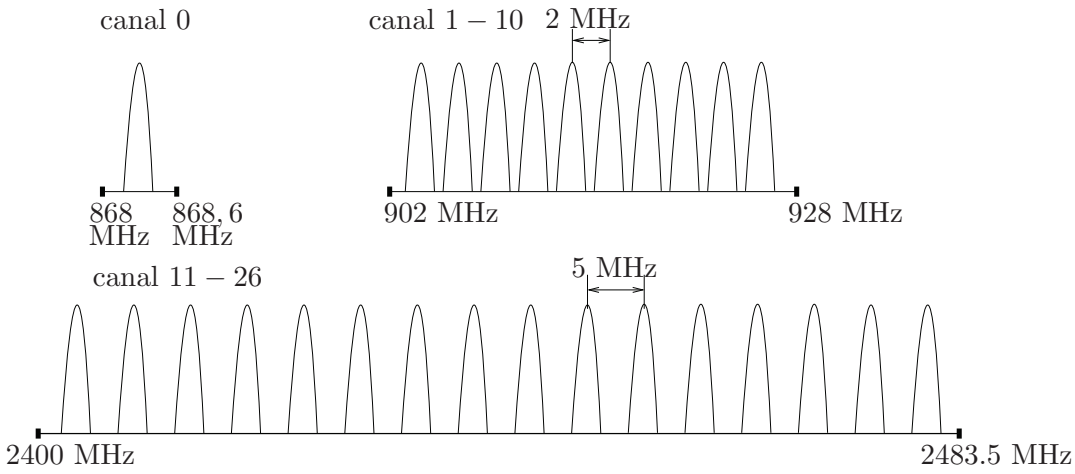


Figure 2.2 – Représentation des canaux disponibles pour le standard IEEE 802.15.4.

Dans la couche PHY, on discerne plusieurs fonctionnalités de gestion du signal et de la ressource radio. D'abord, cette couche peut contrôler et modifier l'état du module radio qui peut se trouver soit en transmission, soit en réception, soit en sommeil. Elle permet également de tester l'occupation du médium durant un court intervalle de temps. Ce test est appelé CCA *(Clear Channel Assessement)*. Il permet de détecter un signal d'une puissance dépassant un seuil fixé et conforme à la modulation de la couche PHY. En outre, la couche PHY permet de définir un indicateur de qualité de lien (LQI : *Link Quality Indicator*) qui mesure le taux d'erreurs binaires dans une trame. Il a pour but d'estimer le taux moyen de pertes de trames entre deux entités du réseau.

Grâce aux fonctionnalités de la couche PHY, la couche MAC peut par exemple mettre en sommeil le module radio dès que possible pour économiser de l'énergie, réaliser des

CCA avant d'envoyer des trames afin d'éviter des collisions, ou encore déterminer un chemin selon la qualité des liens radio entre les nœuds dans le réseau en intégrant le paramètre LQI.

2.1.1.2 Couche MAC IEEE 802.15.4

La couche MAC IEEE 802.15.4 organise l'accès au médium, les opérations liées à la création du réseau et aux associations entre les entités, les différents échanges de trames, et (selon le mode utilisé) la synchronisation par l'envoi de balises appelées *beacons*. Il existe deux modes de fonctionnement de la couche MAC IEEE 802.15.4 : le mode sans suivi de *beacons* et le mode avec suivi de *beacons*. Ces modes sont utilisables selon les besoins applicatifs.

En mode sans suivi de *beacons*, il n'y a pas de synchronisation. Les *beacons* ne servent qu'à la découverte du réseau. Le coordinateur est actif continuellement alors que les feuilles ne sont généralement actives que lorsqu'elles souhaitent communiquer. L'accès au médium n'est pas garanti : les entités sont toujours en concurrence pour émettre.

En mode avec suivi de *beacon*, le coordinateur envoie périodiquement un *beacon* pour synchroniser l'activité des entités. Le *beacon* contient les informations décrivant la structure d'un cycle, c'est-à-dire l'organisation et la durée d'un intervalle de temps (délimité par deux *beacons*). Ce cycle est découpé en deux périodes principales : une période d'activité suivie d'une période d'inactivité. La période d'activité est découpée en seize slots (plages de temps) de durées égales formant une supertrame. La supertrame est composée de trois parties : le *beacon*, la CAP *(Contention Access Period)* et la CFP *(Contention Free Period)*. La CAP définit un intervalle de temps pendant lequel tous les nœuds sont en concurrence pour accéder au médium. Le *beacon* est envoyé par le coordinateur au début du premier slot d'activité de la CAP. La CFP permet, quant à elle, de garantir l'accès au médium par le biais de slots attribués à un couple feuille-coordinateur et appelés GTS *(Guaranteed Time Slots)*. Pendant un GTS, la feuille concernée est la seule à pouvoir communiquer avec le coordinateur. La CFP est optionnelle. Elle peut être soit absente, soit avoir une durée comprise entre un et sept slots. La figure 2.3 décrit la structure d'un cycle global. Durant la période d'inactivité, les entités n'ont rien à faire, elles peuvent passer à l'état sommeil jusqu'au prochain *beacon*.

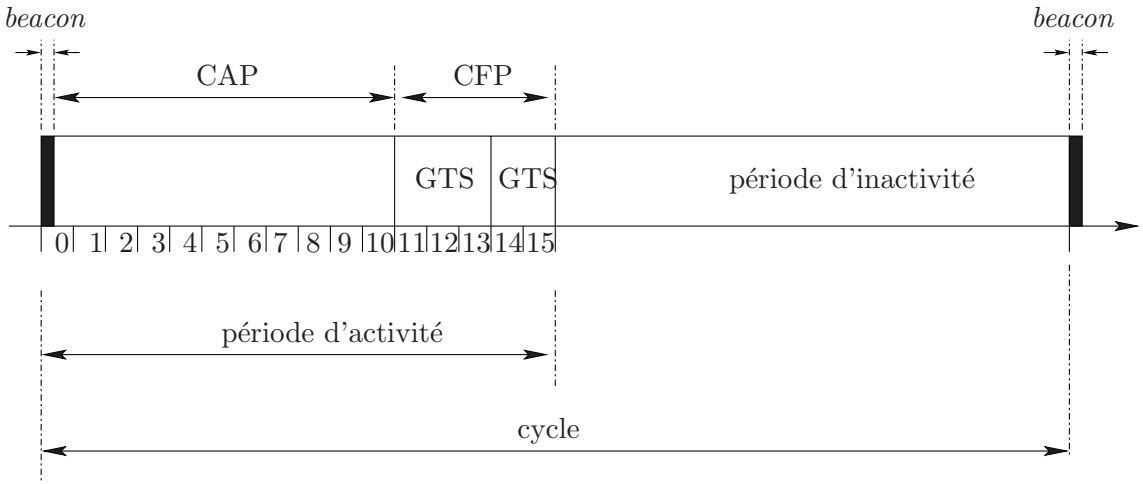


Figure 2.3 – Représentation des différentes périodes d'un cycle global.

La couche MAC IEEE 802.15.4 dispose de plusieurs méthodes d'envoi de trames. Les *beacons* et les acquittements sont toujours envoyés directement. En mode avec suivi de *beacon*, durant la CAP, l'accès au médium est géré selon l'algorithme de CSMA/CA slotté *(Carrier Sense Multiple Access with Collision Avoidance)* et, durant la CFP, les trames envoyées pendant les GTS sont envoyées directement. En mode sans suivi de *beacons*, les trames sont envoyées en CSMA/CA non slotté.

CSMA/CA slotté : Le CSMA/CA slotté est un algorithme basé sur une unité de temps appelée période de *backoff* égale au temps d'envoi effectif de vingt symboles (équivalent à 10 octets) par le module radio, soit 320 μs. Les entités sont synchronisées sur cette échelle de temps grâce à la réception du *beacon*. Ainsi, la supertrame est découpée en slots et les slots sont découpés en périodes de *backoff*. L'intérêt de ce découpage est de réduire le temps de surveillance du médium afin de réaliser une économie d'énergie, point critique dans les réseaux de capteurs sans fil. Le mécanisme de cet algorithme implique trois paramètres :

- NB (nombre de *backoffs*) est le nombre de fois où l'algorithme a été appliqué.
- BE (puissance du *backoff*) est l'exposant qui va influer sur la durée de l'intervalle avant le tirage du prochain *backoff*. Plus il est faible, plus le délai sera court.
- CW *(contention window)* est la fenêtre de contention qui donne le nombre de périodes de *backoff*, temps pendant lequel le médium doit être détecté libre avant de commencer à transmettre la trame.

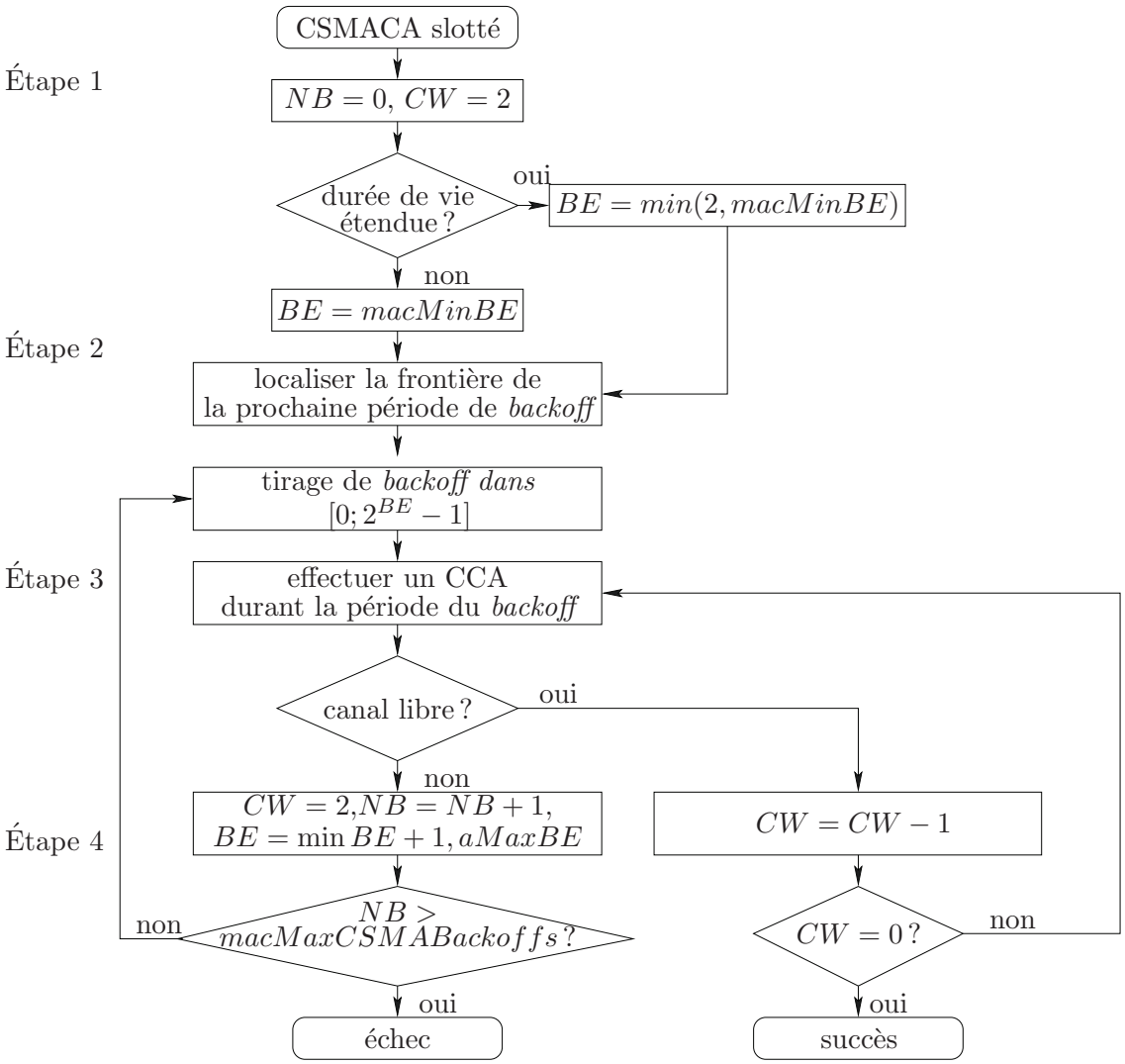


Figure 2.4 – Algorithme du CSMA/CA slotté.

La figure 2.4 présente le diagramme de l'algorithme CSMA/CA slotté. Ce diagramme décrit les phases d'initialisation et les tests effectués pour l'envoi d'une trame. Lors de l'appel de l'algorithme pour l'envoi d'une trame, la couche MAC commence par localiser la frontière de la prochaine période de *backoff* [11]. Puis l'algorithme tire aléatoirement un délai (nombre entier de périodes de *backoff*). Une fois ce délai écoulé, à la frontière de la période de *backoff* suivante, un CCA est effectué. Si le canal est libre, CW est décrémenté. S'il vaut zéro, la couche MAC commande à la couche PHY de transmettre la trame à la prochaine frontière de *backoff*. Si le canal est occupé, BE et NB sont incrémentés, CW reprend pour valeur 2. Si NB atteint *macMaxCSMABackoffs*, l'algorithme renvoie un échec de transmission. Si NB est inférieur à *macMaxCSMABackoffs*, l'algorithme recommence à la phase de localisation de la prochaine frontière de *backoff*. Lors d'un échec de CSMA/CA slotté, la couche MAC effectue *macMaxFrameRetries* tentatives de transmissions de la trame avant de la détruire.

CSMA/CA non slotté : Le CSMA/CA non slotté est proche de l'algorithme de CSMA/CA slotté sans fenêtre de contention et les périodes de *backoff*. Il utilise les périodes de *backoff* comme référence de temps mais, à l'inverse du CSMA/CA slotté, cet algorithme ne nécessite pas une synchronisation des entités entre elles.

La figure 2.5 présente le diagramme de l'algorithme. D'abord les paramètres sont initialisés comme suit : $BE = macMinBE$ et $NB = 0$. Ensuite, un nombre aléatoire de périodes de *backoff* est tiré. Une fois ce délai consommé, un CCA est effectué. Si le canal est libre, alors la couche MAC commence à transmettre. Sinon, NB et BE sont incrémentés (si BE est supérieur à $aMaxBE$, il prend pour valeur $aMaxBE$). Si NB est supérieur à *macMaxCSMABackoffs*, l'algorithme renvoie un échec, sinon, l'algorithme retourne à la phase de tirage de *backoffs*. En cas d'échec, la couche MAC effectue *macMaxFrameRetries* tentatives de transmissions de la trame avant de la détruire.

La couche MAC IEEE 802.15.4 dispose de fonctionnalités de création et de gestion du réseau. Elle fournit quatre types de *scans*, mécanisme qui permet de sonder un ou plusieurs canaux radio, répondant aux besoins de gestion d'un PAN. Le *scan* d'énergie permet d'identifier les canaux les moins encombrés ; il est utilisé par un coordinateur pour

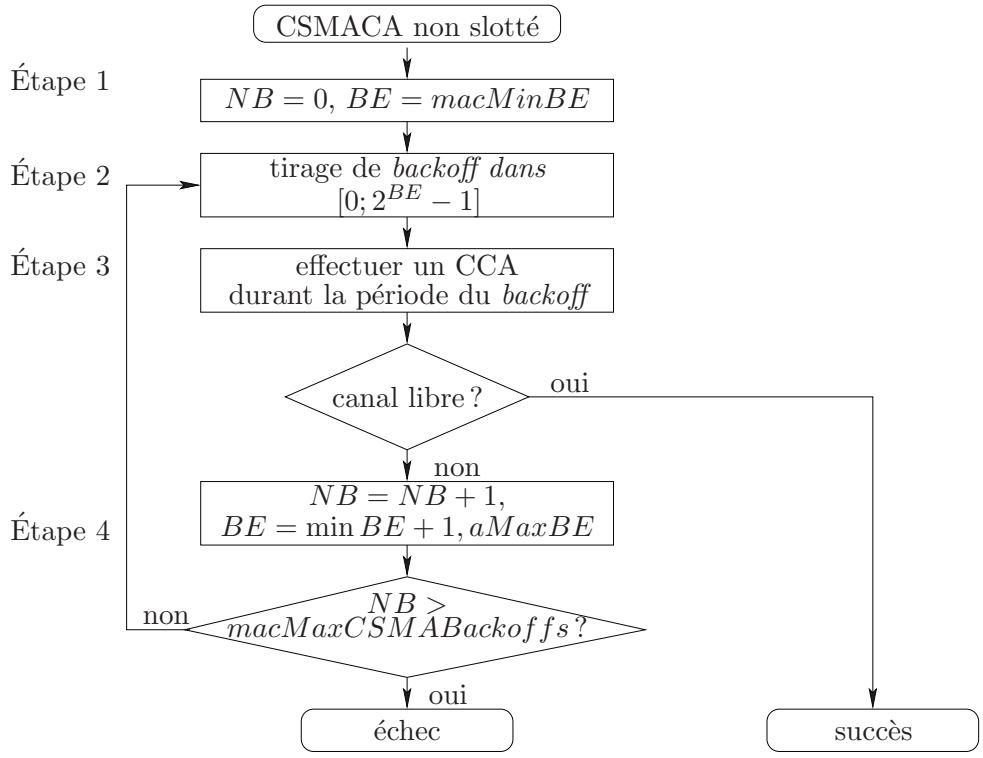


Figure 2.5 – Algorithme du CSMA/CA non slotté.

la création d'un PAN. Le *scan* actif permet, après l'envoi d'une requête de *beacon*, d'établir une liste des identifiants des PAN existants détectés ; il est utilisé par un coordinateur lors de la création de son PAN, afin de sélectionner un identifiant de PAN unique ou par une entité qui cherche à s'associer avec un PAN dont l'identifiant est connu. À la suite d'un *scan* passif, un nœud récupère lui aussi une liste des identifiants des PAN existants, mais par le biais d'une écoute passive pendant une durée donnée. Ce type de scan est utilisé par une entité qui souhaite s'associer à un PAN. Le scan d'orphelin est effectué par une entité qui s'est désynchronisée avec son coordinateur et qui cherche à rétablir sa synchronisation.

À la suite de la création d'un PAN par un coordinateur, une entité à portée peut tenter de s'associer d'abord en effectuant un *scan* (actif ou passif), puis remonte le résultat à la couche supérieure qui sélectionne le PAN et l'adresse du coordinateur auquel s'associer. Une requête d'association est alors émise par l'entité pour le coordinateur. Celui-ci, reçoit la requête et la remonte à la couche supérieure. En mode avec suivi de *beacon*, toutes les entités du PAN sont synchronisées sur le *beacon* du coordinateur du PAN. Une demande de désassociation peut être émise soit par une entité, soit par le coordinateur qui souhaite

désassocier une entité qui lui est reliée. Dans les deux cas, une trame de désassociation est envoyée par le nœud à l'initiative de la requête.

Les trames de données peuvent être échangées de trois manières différentes : en échange direct, en échange indirect et en GTS. L'échange direct a lieu pendant la CAP, en CSMA/CA slotté avec ou sans acquittement. L'échange indirect est initié par les feuilles par le biais d'une requête d'échange. Le coordinateur indique à la feuille s'il a des données en instance (en mode avec suivi de *beacon*, le coordinateur l'indique dans son *beacon*). L'échange des GTS, uniquement disponible en mode avec suivi de *beacon*, est entièrement arbitré par le coordinateur du PAN. Il y a deux types d'allocations : en mode réception (du coordinateur vers l'entité) et en mode émission (de l'entité vers le coordinateur). Une entité demande un GTS au coordinateur du PAN qui lui répond par le biais des descripteurs de GTS contenu dans la trame du *beacon*.

2.1.2 ZigBee

ZigBee [12] désigne un standard défini et soutenu par le ZigBee Alliance [13]. La partie la plus intéressante de ce standard est celle définissant les couches supérieures d'un PAN, en s'appuyant sur IEEE 802.15.4 pour les couches basses. Nous nous intéresserons dans cette partie à la couche réseau. La figure 2.6 permet de schématiser la structure de la pile ainsi constituée.

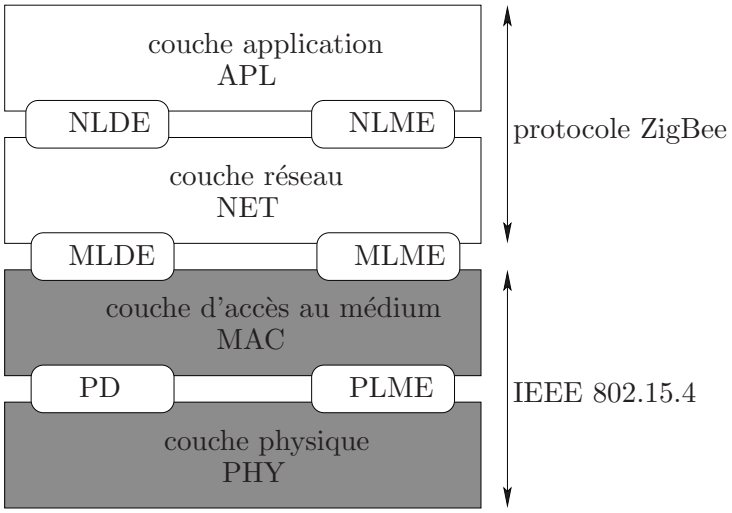


Figure 2.6 – Représentation de la pile IEEE 802.15.4/ZigBee.

Les couches sont reliées entre elles par des interfaces. Ces interfaces sont :

- NLDE pour *Network Layer Data Entity* pour les échanges de primitives de données entre la couche réseau et la couche application,
- NLME pour *Network Layer Management Entity* pour les échanges de primitives de contrôle entre la couche réseau et la couche application,
- MLDE pour *MAC Layer Data Entity* pour les échanges de primitives de données entre la couche MAC et la couche réseau,
- MLME pour *MAC Layer Management Entity* pour les échanges de primitives de contrôle entre la couche MAC et la couche réseau,
- PD pour *PHY Data Layer* pour les échanges de primitives de données entre la couche PHY et la couche MAC,
- PLME pour *PHY Layer Management Entity* pour les échanges de primitives de contrôle entre la couche PHY et la couche MAC.

La couche réseau ZigBee fournit les fonctionnalités liées à la gestion de la topologie, de l'adressage et du routage. Elle assure une compatibilité avec la couche MAC du standard IEEE 802.15.4.

Topologie : La topologie arborescente (ou *cluster-tree*), illustrée dans la figure 2.7, est un cas particulier de la topologie maillée supportée par la couche MAC IEEE 802.15.4. Pour définir les limites de la structure d'un réseau arborescent qui va permettre un routage sans l'usage de tables, l'administrateur du PAN fixe trois paramètres :

- L_m qui définit la profondeur maximale du réseau ($L_m = 1$ pour une topologie en étoile),
- C_m qui définit le nombre maximal de fils par coordinateur,
- R_m qui définit le nombre maximal de fils coordinateur par coordinateur.

Adressage : Chaque entité du réseau possède une adresse MAC unique, longue de 8 octets. Pour réduire la taille des trames échangées, la couche réseau ZigBee attribue une adresse logique dite courte de seulement 2 octets. L'allocation de ces adresses peut être soit aléatoire soit hiérarchique.

Lors de l'allocation d'adresses aléatoires, le coordinateur attribue à chaque entité qui s'associe à lui une adresse au hasard parmi les 65536 disponibles, en excluant celles déjà

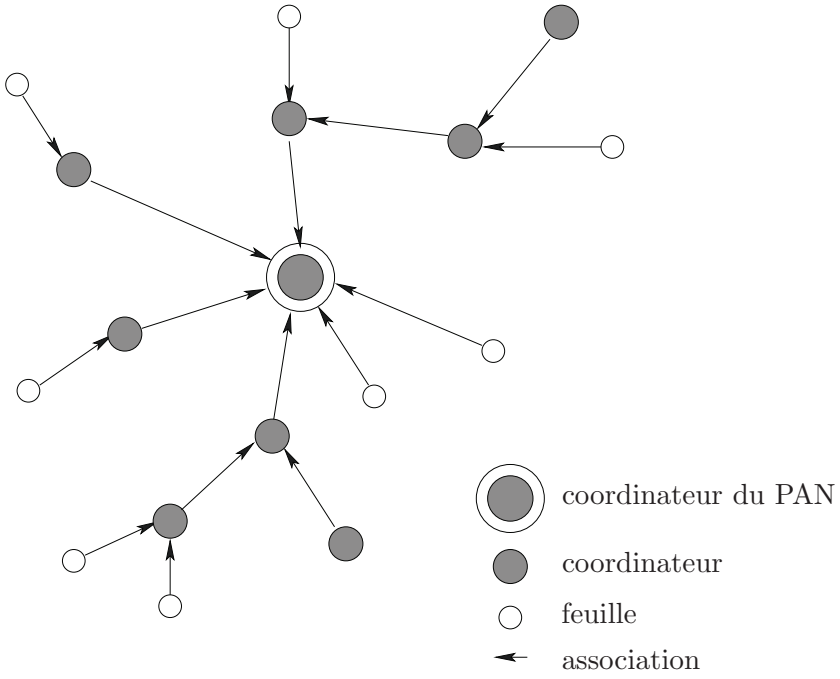


Figure 2.7 – Réseau avec une topologie arborescente.

présentes dans sa table NIB *(Network layer Information Base)*. Les conflits d'adresses sont gérés par un mécanisme supplémentaire [12].

L'allocation d'adresses hiérarchiques est un mécanisme distribué. Chaque coordinateur dispose d'une plage d'adresses déterminée selon sa profondeur dans l'arbre et des paramètres L_m, C_m et R_m [12]. Cette plage est appelée $Cskip$. Elle se calcule selon la profondeur du coordinateur père notée d. La largeur de la plage $Cskip$ est calculée comme suit :

$$Cskip(d) = \begin{cases} 1 + C_m \cdot (L_m - d - 1) & \text{si } R_m = 1, \\ \frac{1 + C_m - R_m - C_m \cdot R_m^{L_m - d - 1}}{1 - R_m} & \text{sinon.} \end{cases}$$

Prenons, par exemple, le cas d'un réseau avec une topologie arborescente associée au jeu de paramètres suivants : $L_m = 3$, $C_m = 3$ et $R_m = 2$. Pour cette configuration, la plage d'adresses $Cskip$ est donnée en fonction de la profondeur d dans le tableau 2.2. À noter qu'un coordinateur de profondeur L_m ne peut accepter aucune association.

Un coordinateur de profondeur p va allouer une adresse et une plage d'adresses aux entités qui vont s'associer à lui selon qu'il s'agit d'un coordinateur ou d'une feuille, ceci en respectant le jeu de paramètres attaché à cet arbre.

Tableau 2.2 – Valeur de $Cskip$ selon la profondeur du nœud pour $L_m = 3$, $C_m = 3$ et $R_m = 2$.

Profondeur du nœud	$Cskip$
0	10
1	4
2	1
3	0

$$Adr_C = Adr_P + 1 + nb_C \times Cskip(d),$$

$$Adr_F = Adr_P + R_m \times Cskip(d) + nb_F.$$

où : Adr_C est l'adresse allouée pour un nouveau coordinateur, Adr_P l'adresse du coordinateur père, nb_C le nombre actuel de coordinateurs fils.

Adr_F est l'adresse allouée pour une nouvelle feuille et nb_F le nombre actuel de feuilles du coordinateur.

La figure 2.8 présente un arbre dont la configuration est définie par les paramètres $L_m = 3$, $C_m = 3$, $R_m = 2$, et indique les adresses obtenues pour chaque entité.

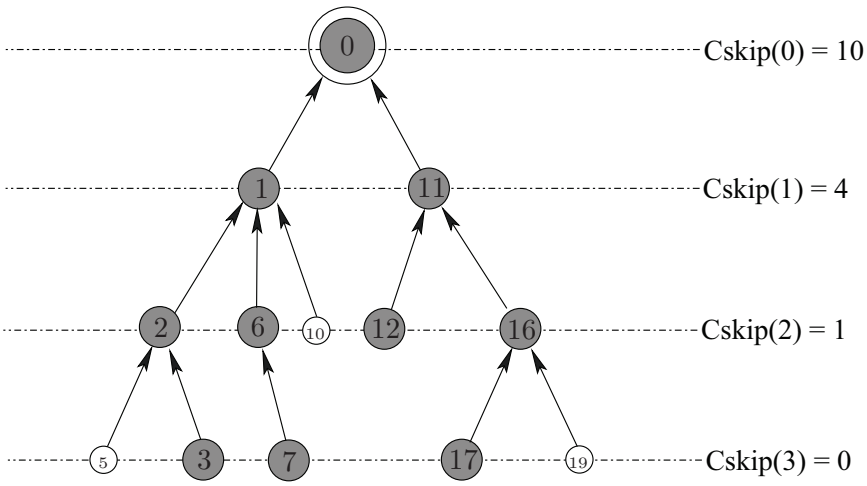


Figure 2.8 – Exemple de topologie arborescente et de l'adressage hiérarchique.

Routage : Le routage est réalisé par les coordinateurs. Chacun d'entre eux peut maintenir une table de routage afin de déterminer le prochain saut d'une trame dont il n'est pas le destinataire final. Seul le routage hiérarchique, correspondant à l'adressage hiérarchique, est abordé dans ce mémoire car il est utilisé par MaCARI et qu'il est plus économe en

Algorithme 1 Algorithme de routage hiérarchique

si $Adr < DestFinale \leq Adr + Cskip(d-1)$ **alors**
  si $DestFinale \geq Adr + R_m \times Cskip(d)$ **alors**
    $Dest \leftarrow DestFinale$
  sinon
    $Dest \leftarrow Adr + 1 + \lfloor \frac{DestFinale-(Adr+1)}{Cskip(d)} \rfloor \times Cskip(d)$
  fin si
sinon
  $Dest \leftarrow Adr_P$
fin si

trafic qu'AODV [14] *(Ad hoc On-Demand Distance Vector)*, proposé dans la norme pour l'adressage aléatoire (ou pour réduire le nombre de sauts).

L'adressage hiérarchique permet la mise en œuvre du routage hiérarchique. Ce type de routage évite la construction de tables de liens et la maintenance associée (comme les envois de trames pour vérifier l'état des liens). Le cheminement des trames suit l'arbre. Il existe trois possibilités concernant ce mode de routage : soit la trame est routée vers le haut de l'arbre, soit vers le bas de l'arbre via un coordinateur fils, soit vers une feuille du coordinateur. Pour déterminer le cheminement d'une trame, un coordinateur applique un algorithme donnant l'adresse du prochain saut selon sa propre adresse et l'adresse de la destination.

Soit Adr l'adresse du coordinateur qui reçoit une trame à router avec le routage hiérarchique, d sa profondeur, Adr_P l'adresse de son père, $DestFinale$ l'adresse de la destination finale de la trame et $Dest$ l'adresse du prochain saut.

Le coordinateur Adr applique l'algorithme 1. Le coordinateur teste tout d'abord si la destination finale est son descendant. Si ce n'est pas le cas, la trame est routée vers le père du coordinateur. Si c'est un descendant, le nœud teste si la destination finale est une de ses feuilles : si c'est le cas, il route la trame vers sa feuille. Sinon, il route la trame vers le coordinateur fils dont l'adresse est déterminée comme suit : le coordinateur cherche quel fils possède la plage qui contient l'adresse de la destination finale. Comme chaque fils possède une plage de $Cskip(d)$ adresses, et que l'adresse du premier fils est $Adr+1$, le père calcule le numéro du fils grâce à $x = \lfloor \frac{DestFinale-(Adr+1))}{Cskip(d)} \rfloor$. Puis, avec le numéro x du fils, le nœud recalcule l'adresse de son x-ème coordinateur fils avec $Adr+1+x \times Cskip(d)$.

2.1.3 Solutions alternatives

Les solutions alternatives à la norme IEEE 802.15.4 qui présentent une maturité suffisante sont principalement WirelessHART et ISA 100.11a. Dans cette partie, nous décrivons les principales caractéristiques de chacun de ces protocoles puis nous citons certains protocoles non standardisés.

2.1.3.1 WirelessHART

Le standard WirelessHART [15] est destiné à des applications principalement industrielles dont l'objectif est d'effectuer des contrôles et des relevés de mesures. C'est un standard ouvert qui offre une bonne fiabilité et une approche pour économiser l'énergie. Le protocole de communication WirelessHART est développé par la fondation HART Communication. La Commission Électrotechnique Internationale (IEC) a validé les spécifications de WirelessHART comme un standard international complet en mars 2010. Nous allons tout d'abord aborder les topologies disponibles, puis détailler les couches basses de la pile protocolaire (à savoir les couches physique, liaison de données et réseau).

WirelessHART supporte les topologies maillées, en étoile ou une combinaison des deux. Toutefois, le standard est généralement utilisé en mode maillé. Un tel réseau est composé de trois principaux types d'équipements : des entités autonomes sans fils (WFD : *Wireless Field Devices*), une passerelle et un gestionnaire du réseau (NM : *Network Manager*). Les entités ont toutes des caractéristiques identiques et disposent toutes de fonctionnalités de routage ; il n'y a pas de RFD comme dans ZigBee. Une passerelle, alimentée sur le secteur ou autonome en énergie, permet de garantir la synchronisation des entités et d'établir un lien avec un réseau externe. Le gestionnaire du réseau organise entièrement le réseau (il gère les mécanismes d'association et de désassociation) et ordonnance l'activité des WFD. La figure 2.9 propose un exemple de réseau WirelessHART dans laquelle le NM est intégré dans la passerelle.

La couche PHY de WirelessHART est la même que celle du standard IEEE 802.15.4 dans la bande de fréquences 2.4 GHz, avec une modulation O-QPSK offrant un débit de 250 kbps pour chacun des 16 canaux disponibles.

La couche liaison de données fournit des fonctionnalités de fiabilité, de gestion de

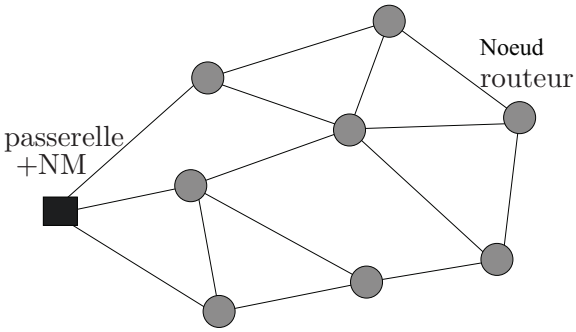


Figure 2.9 – Exemple d'un réseau WirelessHART maillé.

l'accès au médium par les entités du réseau et de sécurité. La synchronisation des entités est basée sur le temps de la passerelle. Le temps est découpé en slots de 10 ms. L'accès au médium se réalise en TDMA *(Time Division Multiple Access)* avec sauts de canal [15]. Ce type de méthode d'accès permet de garantir le déterminisme : l'envoi d'une trame est effectué dans un délai borné et sans collision. La robustesse des transmissions est améliorée grâce à la possibilité d'interdire l'émission sur un ou plusieurs canaux déjà occupés (ou pollués) pour éviter les perturbations. Une série de slots définit une supertrame TDMA. Une supertrame est périodique. Les échanges sont toujours acquittés et chaque acquittement contient des informations temporelles permettant de continuellement synchroniser les opérations en TDMA dans le réseau.

La couche réseau fournit deux types de routage supportés par l'ensemble des entités : le routage centralisé par état de liens et le routage par la source. Le routage centralisé utilise la notion de graphes. Un graphe regroupe une collection de chemins *via* les nœuds du réseau. Chaque entité maintient une table de graphes qui possèdent chacun un identifiant. Les chemins dans chaque graphe sont créés par le NM et transmis à chacune des entités du réseau. Pour envoyer une trame, le nœud expéditeur indique l'identifiant du graphe selon la destination dans l'entête réseau de la trame. Chaque entité qui reçoit une trame consulte dans sa table le graphe correspondant à l'identifiant du graphe contenu dans l'entête de la trame. Par cette méthode, le nœud détermine ensuite l'adresse du prochain destinataire et lui transmet la trame. Les nœuds suivants réalisent le même processus jusqu'au destinataire final. Le routage par la source, à la différence du routage centralisé, ne nécessite pas que les nœuds intermédiaires possèdent une table de graphes. Pour envoyer une trame à sa destination, l'expéditeur (la source) inscrit de manière séquentielle la liste

de toutes les adresses des nœuds par lesquels la trame doit être routée. Chaque nœud n'a plus qu'à suivre la route contenue dans l'entête de la trame. Ces méthodes de routage permettent d'avoir un contrôle sur le trafic ce qui ajoute de la robustesse au réseau mais consomme une partie du slot TDMA. Les informations liées au réseau sont régulièrement remontées et utilisées à des fins statistiques ; elles rapportent les découvertes de nouvelles entités, les déconnexions et les désassociations des entités. Cependant, le type de routage utilisé dans WirelessHART n'inclue pas de stratégie d'économie d'énergie.

2.1.3.2 ISA 100.11a

Le standard ISA 100.11a [2] est dédié aux réseaux industriels bas débit pour les applications non critiques de surveillance et de contrôle. Il a de nombreux points communs avec le standard WirelessHART. C'est un standard ouvert basé sur une basse consommation énergétique, une bonne évolutivité et fiabilité, et sur la gestion d'un très grand nombre d'entités. Ces entités autonomes sont organisées en sous-réseaux et reliées par le biais d'une passerelle à des équipements de gestion alimentés en permanence formant l'épine dorsale du réseau *(backbone)*. L'interconnexion de plusieurs sous-réseaux forme un réseau ISA 100.11a. Nous allons détailler les différents dispositifs et topologies, puis les couches basses de la pile protocolaire du standard.

Le standard propose deux types de topologies : en étoile et maillée. Il existe deux types d'entités autonomes sans fil, tout comme dans ZigBee ; des FFD qui sont capables de router et des RFD qui ne le peuvent pas. Dans les types d'équipements alimentés d'un réseau ISA 100.11a existent : la passerelle, le gestionnaire de système et de sécurité, et les routeurs pivots. La passerelle et le gestionnaire de système (NM) fonctionnent de manière très similaire aux équipements équivalents dans le standard WirelessHART. Les routeurs pivots sont nécessaires dès que les réseaux sont étendus. La figure 2.10 illustre un exemple de réseau ISA 100.11a avec des routeurs pivots. Les liens entre les différentes entités, autonomes ou alimentées, représentent le type de liaison : soit point à point (Route 1) ou multipoints (Route 2..N).

La couche physique d'ISA 100.11a est la même que celle du standard IEEE 802.15.4 dans la bande de fréquences 2.4 GHz.

La couche liaison de données repose sur une partie de la couche MAC IEEE 802.15.4.

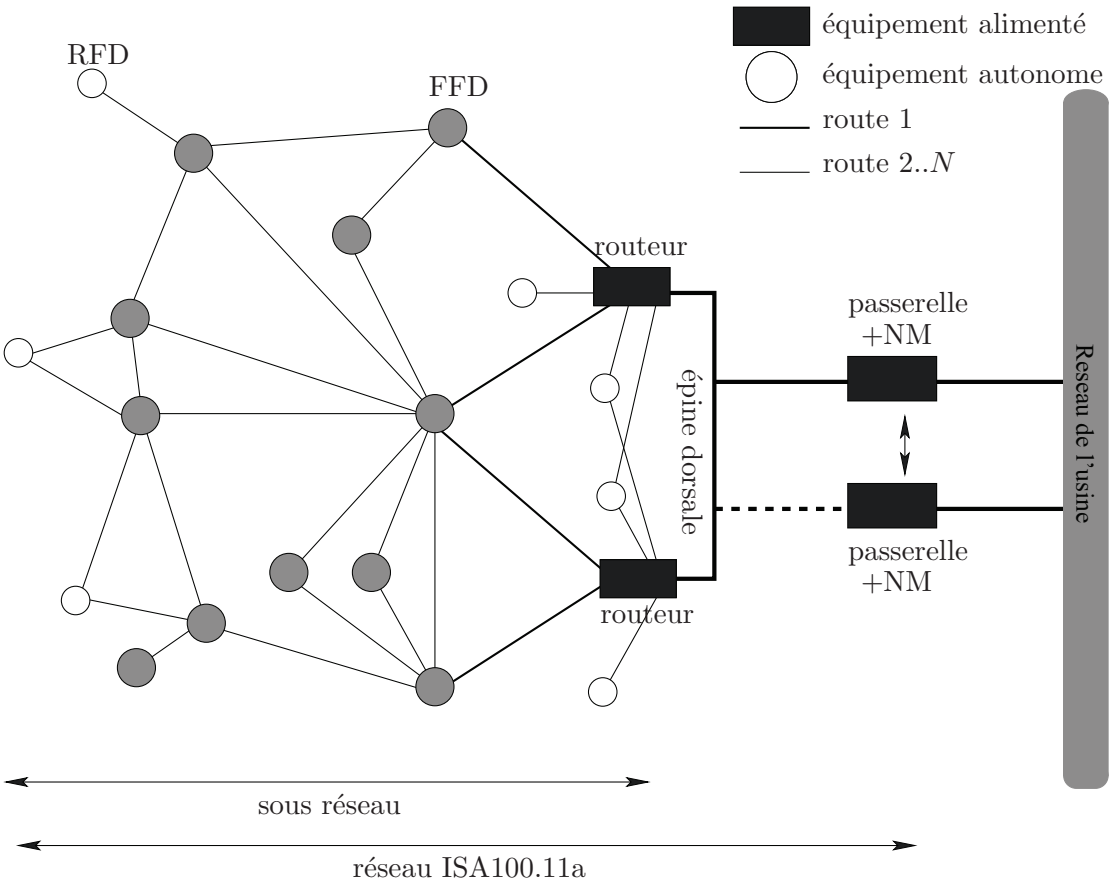


Figure 2.10 – Exemple d'un réseau ISA100 connecté au réseau d'une usine.

Cette couche partage aussi beaucoup de similitudes avec WirelessHART : notamment la synchronisation et les échanges en TDMA avec sauts de fréquence sont très proches de ceux de WirelessHART. Sur ce point, dans un réseau l'ISA 100.11a, le TDMA peut être combiné avec le CSMA/CA pour augmenter la disponibilité et accélérer la formation et la maintenance du réseau en introduisant des périodes de contention (CAP). Par ce mécanisme, on peut introduire plusieurs classes de trafic : le trafic purement TDMA, le trafic purement CSMA/CA ou le trafic hybride.

La couche réseau utilise un routage centralisé par état de liens. Elle fournit les services d'adressage et une qualité de service. L'adressage des entités est fait sur 128 bits : l'adresse de l'entité est représentée sur 64 bits et les sous-réseaux sur 64 bits. Cela permet d'adresser un nombre important de dispositifs. Les méthodes de routage utilisées dans ISA 100.11a n'incluent pas de stratégie d'économie d'énergie.

2.1.3.3 Protocoles non standardisés

Il existe d'autres protocoles MAC non standardisés utilisés pour les réseaux de capteurs dont les propriétés visent à économiser de l'énergie. Bien que dans le bilan de cette partie nous nous concentrerons sur les protocoles standardisés, nous pouvons citer quelques uns de ces protocoles pouvant être répartis en trois groupes :

- les protocoles basés sur un séquencement temporel comme E-MAC [16], utilisant le découpage de slots TDMA,
- les protocoles basés sur la contention comme B-MAC [17], utilisant le LPL*(low power listening)*,
- les protocoles hybrides comme Z-MAC [18], alternant CSMA/CA et TDMA, ou Eco-MAC [19], combinant contention, découpage temporel et sauts de fréquence.

2.1.4 Bilan

L'analyse des standards réalisée dans cette partie permet de dresser un comparatif des fonctionnalités à différents niveaux. Le tableau 2.3 regroupe par couche les propriétés de chacun de ces standards afin d'établir un bilan de leurs caractéristiques.

Si les trois standards ont la même couche physique, ils ont des approches différentes

Tableau 2.3 – Comparatif entre les standards.

Caractéristiques	**ZigBee**	**WirelessHART**	**Isa 100.11a**
Couche réseau	Routage hiérarchique ou AODV	Routage centralisé (par état de liens)	Routage centralisé (par état de liens)
Couche MAC	CSMA/CA et GTS	TDMA avec saut de fréquences	TDMA avec sauts de fréquences + CSMA/CA
Couche physique	IEEE 802.15.4	IEEE 802.15.4	IEEE 802.15.4

notamment au niveau de la méthode d'accès et du routage. Ces approches ont un impact fort sur la consommation énergétique des entités de ces réseaux.

Méthode d'accès : Pour un réseau WirelessHART et ISA 100.11a, le saut de fréquence permet une certaine robustesse et une parallélisation des transmissions. Un réseau ZigBee, dont la création est précédée d'un *scan* initial des canaux, n'utilise pour fonctionner qu'un seul et unique canal. Cela permet un contrôle précis de la ressource radio et de son utilisation, la cohabitation avec d'autres réseaux et l'indépendance physique des réseaux.

ISA 100.11a et WirelessHART sont des standards basés sur des échanges en TDMA, associés à la présence d'un équipement, généralement alimenté, qui gère l'activité des nœuds. Le TDMA est une méthode garantissant un déterminisme nœud à nœud au niveau MAC. ZigBee, en revanche utilise une méthode d'accès hybride ; il offre, en mode avec suivi de *beacon*, un déterminisme limité *via* les GTS dans une topologie maillée. Le CSMA/CA offre une meilleure réactivité et une gestion de l'accès au médium dynamique et décentralisée. Cependant, on remarque que ces standards ne proposent pas de mécanisme garantissant un déterminisme associé à un délai de bout en bout borné (mis à part WirelessHART au prix d'une configuration fortement statique).

Routage : Au niveau de la couche réseau, ISA 100.11a et WirelessHART utilisent le routage centralisé et le routage par la source. Ces méthodes de routage permettent de renforcer la fiabilité du réseau en permettant l'exploitation de chemins multiples d'une source vers sa destination. En contrepartie, cette centralisation impose de nombreuses transmissions de messages de contrôle et d'entretien des tables de routage (transmissions entre les entités et le gestionnaire du réseau), ce qui entraine une baisse de la réactivité

et un flux de trafic de contrôle élevé. Un réseau arborescent ZigBee peut utiliser le routage hiérarchique qui est une méthode décentralisée permettant un routage statique sans échange de données de contrôle. ZigBee peut également utiliser le routage AODV qui établit les routes à la demande : il a l'avantage de ne générer du trafic de contrôle que lors des transmissions mais, dès lors, il est susceptible de générer beaucoup de surcharge.

Économie d'énergie : L'économie d'énergie est réalisée au niveau MAC quand le nœud dort. Pour WirelessHART, tous les nœuds étant des routeurs, l'ensemble des entités autonomes sont susceptibles d'être sollicitées. Dans ZigBee et ISA 100.11a, qui spécifient l'utilisation de RFD, ce n'est pas le cas. De plus, le routage employé dans WirelessHART et ISA 100.11a n'offre pas de stratégie d'économie d'énergie. Dans ZigBee, les coordinateurs sont davantage sollicités par rapport aux RFD et bornent la durée de vie du réseau. Néanmoins, à l'inverse des autres standards, ZigBee n'a pas besoin d'une infrastructure alimentée en permanence ; toutes les entités du réseau économisent de l'énergie à l'exception du coordinateur du PAN.

Chacun de ces standards possède des caractéristiques propres qui peuvent constituer des limites incompatibles (et/ou des contraintes) avec une partie des demandes des industriels. La plupart des applications industrielles nécessite que les paramètres d'une pile protocolaire soient largement paramétrables selon les exigences de latence, de différenciation de service, de déterminisme et d'autonomie.

En cela, le projet OCARI a permis de spécifier une solution complémentaire qui propose la méthode d'accès MaCARI combinant flexibilité, déterminisme et économie d'énergie.

2.2 MaCARI

Le protocole MaCARI (MAC pour OCARI) [20] est un protocole MAC basé sur le standard IEEE 802.15.4 ; il repose sur la couche PHY de ce standard et propose des variantes à sa couche MAC. Il reprend le mode d'adressage et le routage hiérarchique de ZigBee. MaCARI a été conçu pour permettre :

– l'économie d'énergie de tous les éléments du réseau,

– la garantie d'un délai de bout en bout borné,

– "l'étanchéité" du réseau (pas d'interférence entre deux réseaux OCARI),

– une maintenance simple du réseau (en arbre) afin de réduire le trafic de contrôle,

– une différenciation de service au niveau MAC.

La description de MaCARI qui suit est en partie inspirée de la thèse de G. Chalhoub [8].

Dans un réseau OCARI, trois types de nœuds sont présents : un coordinateur du PAN, des coordinateurs et des feuilles. Le coordinateur du PAN, appelé CPAN par la suite, est le nœud qui crée le réseau avec une topologie arborescente, initie et maintient une synchronisation entre l'ensemble des nœuds du réseau. Il joue souvent le rôle de puits de données ou de passerelle pour s'interconnecter à d'autres réseaux. Les coordinateurs sont des nœuds qui contribuent au routage des informations et participent à l'étape de synchronisation du réseau. Les feuilles sont les capteurs et/ou les actionneurs ayant un rôle applicatif. Les feuilles sont attachées à un coordinateur et communiquent uniquement avec lui. Un coordinateur et l'ensemble des feuilles qui lui sont associées constituent une étoile. Un exemple de topologie de réseau est présenté sur la figure 2.11.

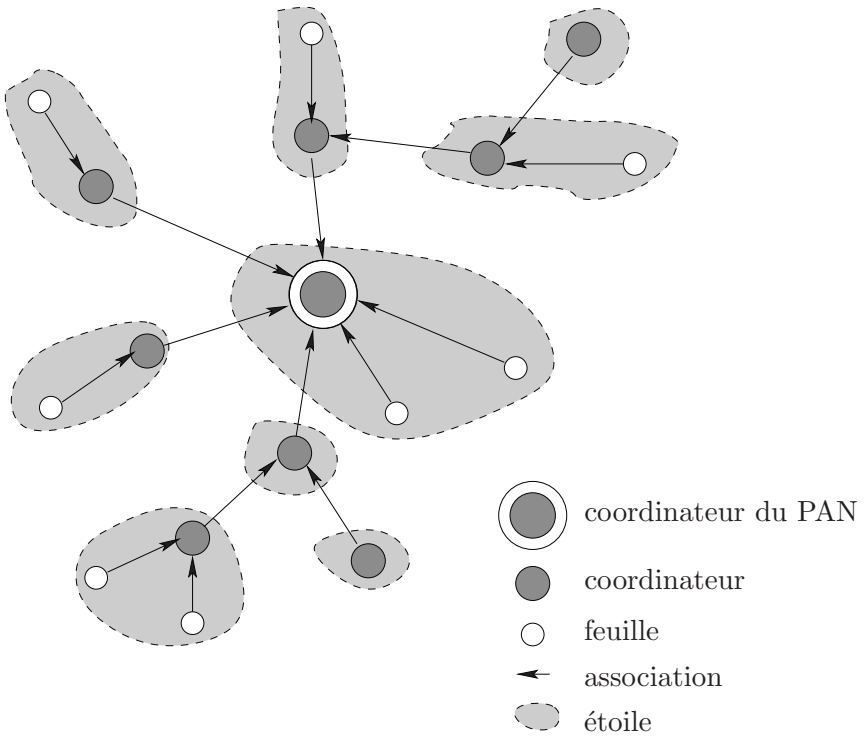


Figure 2.11 – Exemple de topologie d'un réseau OCARI.

La couche MaCARI fournit les fonctionnalités suivantes :

- la gestion de la synchronisation et le découpage temporel des périodes d'activité du réseau,
- l'accès au médium,
- la création du réseau et la gestion de sa topologie,
- l'affectation des adresses hiérarchiques,
- le relais garanti de trames pour le trafic de type temps contraint,
- l'estampillage temporel des trames,
- l'acheminement de trames de bout en bout en quatre cycles au maximum (quelles que soient la source et la destination de l'échange).

2.2.1 Synchronisation et découpage temporel

MaCARI divise le temps en trois périodes principales qui forment un cycle global :

- une période de synchronisation $[T_0; T_1]$,
- une période d'activité $[T_1; T_3]$ (elle-même sous-découpée en $[T_1; T_2]$ et $[T_2; T_3]$),
- une période d'inactivité $[T_3; T_0]$.

Ces périodes sont présentées sur la figure 2.12.

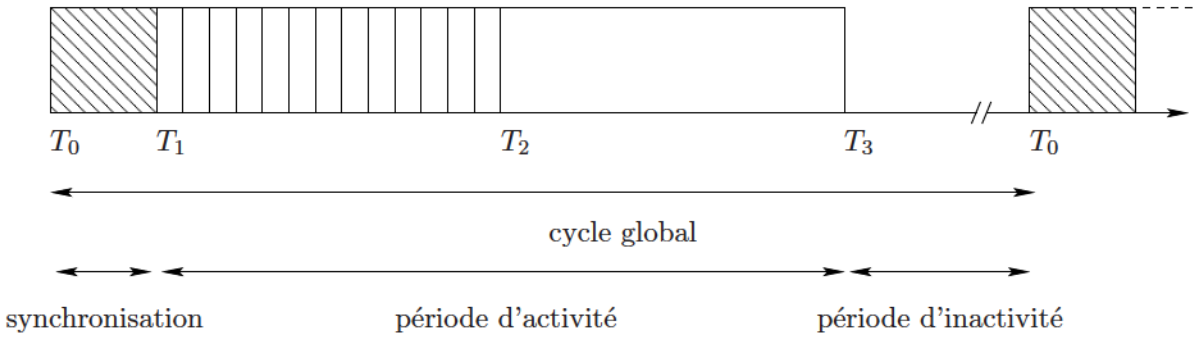


Figure 2.12 – Le cycle global de MaCARI.

Durant la période de synchronisation $[T_0; T_1]$, le CPAN diffuse un *beacon* de synchronisation qui précise le dimensionnement du cycle global ainsi que le sous-découpage de la période d'activité. Pour pouvoir atteindre toutes les entités du réseau qui ne sont pas forcément à portée du CPAN, les coordinateurs propagent le *beacon* à tour de rôle [21, 22]. La figure 2.13 décrit le principe de cascade de *beacons* pour un réseau composé de sept

coordinateurs. Cette organisation permet d'éviter toute collision de *beacons*. Pour simplifier, les feuilles ne sont pas représentées sur cette topologie, mais nous considérons qu'à chaque coordinateur est associé un ensemble de feuilles. Sur la figure, a_i, b_i, c_i, d_i, e_i, f_i et g_i désignent respectivement l'ensemble des feuilles associées aux coordinateurs A, B, C, D, E, F et G. Nous considérons dans cet exemple que A est le CPAN et qu'il adopte l'ordre de synchronisation (correspondant à un parcours en largeur) suivant : (B, C, D, E, F, G).

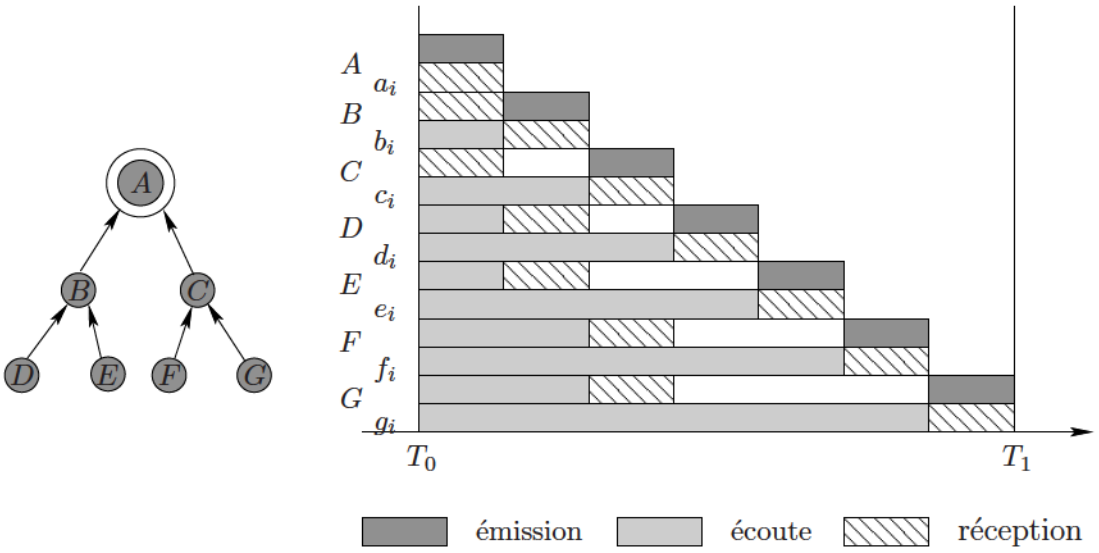


Figure 2.13 – Principe de la cascade de *beacon* dans MaCARI.

A commence la cascade en diffusant le premier *beacon*. Ce *beacon* est reçu (au moins) par ses coordinateurs fils, B et C, et par ses feuilles a_i. Grâce à l'ordre de synchronisation contenu dans le *beacon*, B sait qu'il est le coordinateur suivant à envoyer le *beacon*. C sait qu'il doit attendre l'envoi de B. B envoie donc son *beacon* en deuxième, suivi par C. De même, quand D et E reçoivent le *beacon* de B, D sait qu'il doit attendre l'envoi de C, et E sait qu'il doit attendre l'envoi de C et l'envoi de D. Ainsi grâce à l'ordre de synchronisation qui est diffusé dans le *beacon*, les coordinateurs savent à quel instant il faut envoyer le *beacon* qu'ils ont reçu, sans générer de collisions. Une fois que tous les coordinateurs ont envoyé leur *beacon*, toutes les entités du réseau (feuilles et coordinateurs) sont synchronisées sur T_1. Une version optimisée, prenant en compte le temps de traitement pour réduire la durée de cette période de synchronisation et d'économiser de l'énergie, est décrite dans [8].

La période d'activité est décomposée en deux parties : une période de séquencement d'activités et de relais, et une période de routage. Le séquencement d'activités et le relais se font dans l'espace temporel $[T_1; T_2]$ durant lequel chaque étoile se voit allouer un intervalle de temps propre. Durant cet intervalle, le coordinateur de l'étoile communique avec les feuilles qui lui sont associées afin de collecter les données produites par les capteurs et de transmettre des ordres aux actionneurs. Nous appelons cet intervalle de temps la période d'activité d'une étoile. Il permet à chaque étoile d'organiser son activité sans craindre des collisions ou des interférences dues à l'activité d'autres étoiles en un même lieu. Ainsi, un coordinateur est capable d'allouer un GTS (selon la norme IEEE 802.15.4) sans craindre que les slots soient utilisés par le coordinateur d'une autre étoile. Un autre intervalle de temps supplémentaire est alloué à chaque coordinateur (à l'exception du CPAN) afin qu'il communique avec son père. Nous appelons cet intervalle de temps l'intervalle de relais. Ces intervalles de relais sont prévus pour les échanges protégés de trames qui doivent être envoyées sans risque de collisions (trafic de type temps contraint). Ce sont des intervalles de temps à accès garanti étant donné que le coordinateur et son père sont les deux seules entités actives dans le réseau, durant cet intervalle de temps. Ce découpage temporel est représenté sur la figure 2.14. Dans le projet OCARI, la solution décrite pour les échanges coordinateur-feuilles a été plus spécifiquement développée par le partenaire LATTIS.

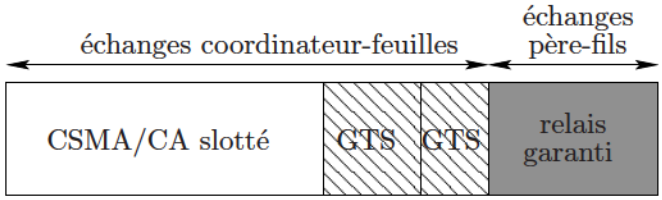


Figure 2.14 – Période d'activité suivie par un intervalle de relais.

À T_2, toutes les étoiles ont eu une période d'activité qui a permis aux coordinateurs de récolter le trafic généré par leurs feuilles et de relayer la partie prioritaire de ce trafic à leur père ou à leurs fils. L'acheminement de la partie du trafic non-prioritaire se fait dans la période d'activité non ordonnancée $[T_2; T_3]$. Seuls les coordinateurs sont actifs durant cette période tandis que toutes les feuilles dorment et économisent de l'énergie. Les coordinateurs communiquent entre eux selon un protocole de routage hiérarchique

amélioré grâce à la connaissance de la table de voisinage des nœuds et de leur énergie résiduelle (par l'utilisation d'EOLSR pour *Enhanced Optimized Link State Routing* [23], version améliorée de OLSR [24]).

Durant la période d'inactivité, toutes les entités du réseau économisent de l'énergie en se mettant en mode sommeil. La durée de cette période dépend de la réactivité exigée par l'application. Le défi lié à cette période est le compromis entre l'économie d'énergie et le délai (que ce soit le délai d'accès au médium ou le délai de bout-en-bout). Plus cette période est longue, plus les entités économisent de l'énergie, mais plus le délai introduit est grand.

L'activité d'un nœud dans le réseau dépend de son type. Un coordinateur est actif durant la période $[T_0; T_1]$, de T_0 jusqu'à ce qu'il envoie son *beacon*, alors qu'une feuille est active de T_0 jusqu'à ce qu'elle reçoive un *beacon*. Dans $[T_1; T_2]$, un coordinateur est actif durant la période de collecte de son étoile, durant l'intervalle de relais avec son père et durant les intervalles de relais de ses fils. Une feuille est active uniquement durant la période d'activité de son étoile. Dans $[T_2; T_3]$, tous les coordinateurs sont actifs pour acheminer les trames à leur destination finale, les feuilles ne sont pas actives. Finalement, aucune entité du réseau n'est active durant $[T_3; T_0]$.

2.2.2 Accès au médium

Dans MaCARI, les *beacons* dans $[T_0; T_1]$ sont envoyés directement (le canal est supposé disponible). Les échanges intra-étoile sont gérés en CSMA/CA slotté pour le trafic de type non-prioritaire, et sans CSMA/CA durant les intervalles de GTS pour le trafic de type prioritaire. Les échanges durant les intervalles de relais garanti sont effectués sans CSMA/CA. Durant la période $[T_2; T_3]$, l'accès est géré selon le protocole CSMA/CA slotté.

2.2.3 Création du réseau et gestion de la topologie

Le CPAN est le nœud responsable de la création du réseau. Il choisit un identifiant pour le PAN (nommé PAN ID) et diffuse un *beacon* périodiquement. Les entités qui reçoivent ce *beacon* peuvent envoyer une requête d'association au CPAN pour rejoindre

le réseau, soit en tant que coordinateur, soit en tant que feuille. Une fois accepté en tant que coordinateur, le nouveau nœud commence à propager le *beacon* et permet à d'autres entités de rejoindre le réseau. Après chaque association d'un nouveau coordinateur, une primitive est remontée au CPAN pour l'informer de cette nouvelle association. Ainsi, une topologie arborescente est créée entre les différents nœuds du réseau.

2.2.4 Affectation des adresses hiérarchiques

Dans un réseau de topologie arborescente, le coordinateur du PAN transmet dans le *beacon* trois paramètres qui définissent la topologie en arbre du réseau. Comme vu dans la partie 2.1.1, ces trois paramètres sont la profondeur maximale du réseau L_m, le nombre maximal de fils par coordinateur C_m, le nombre maximal de fils coordinateurs par coordinateur R_m.

Le CPAN possède toujours l'adresse 0. Au moment de l'association, à chaque nœud est affectée une adresse hiérarchique. L'allocation des adresses se fait en se basant sur la profondeur du coordinateur et sur les trois paramètres L_m, R_m et C_m en utilisant les mêmes formules que ZigBee. Cela permet à MaCARI d'appliquer le protocole de routage hiérarchique décrit précédemment.

2.2.5 Relais garanti pour une différenciation de service

Un relais de trames de niveau MAC est effectué par MaCARI, pour faire cheminer le trafic de type prioritaire. Ayant connaissance de l'adresse hiérarchique de la destination finale et des paramètres de la topologie, MaCARI réalise ce relais en suivant l'arbre et en appliquant le protocole de routage hiérarchique durant les intervalles de relais garanti entre père et fils, ceci sans solliciter la couche réseau des entités intermédiaires.

C'est par ce mécanisme que MaCARI remplit son objectif d'offrir deux niveaux de qualité de service : un accès garanti pour le trafic prioritaire et un accès non-garanti pour le trafic non-prioritaire. L'accès garanti est réalisé durant les intervalles de relais garanti entre les couples de coordinateurs père-fils, et durant les GTS réservés aux communications entre le coordinateur et ses feuilles. L'accès non-garanti est réalisé durant les périodes d'activité des étoiles et durant $[T_2; T_3]$ en utilisant l'algorithme de CSMA/CA slotté.

2.2.6 Estampillage temporel des trames

Grâce à la synchronisation sur l'instant T_1 commun à toutes les entités du réseau, MaCARI possède une connaissance temporelle qui lui permet de fournir une notion de temps commune à toutes les entités. Ce temps peut servir à mettre en œuvre un mécanisme d'estampillage pour les trames de données. Il inclut un numéro de cycle global, et une durée depuis le dernier T_1.

2.2.7 Exemple de cycle global

La figure 2.15 met en évidence le découpage d'un cycle MaCARI pour un arbre binaire de 7 nœuds qui favorise le trafic ascendant. En effet, pendant $[T_1; T_2]$, les étoiles les plus profondes dans l'arbre ont leur période d'activité en premier. Pendant ces périodes d'activité, le coordinateur de chaque étoile échange des données en CSMA/CA slotté et/ou en GTS avec ses feuilles puis remonte le trafic garanti à son coordinateur père. Ensuite, pendant $[T_2; T_3]$, les coordinateurs échangent le trafic non garanti. Enfin, tous les nœuds entrent en période d'inactivité jusqu'au début du prochain cycle.

2.2.8 Acheminement de bout en bout en moins de quatre cycles

L'acheminement de bout en bout pour le trafic prioritaire est réalisé *via* les intervalles de relais par l'organisation du séquencement des périodes d'activité des étoiles de l'arbre.

L'exemple vu dans la figure 2.15 présente un séquencement de l'activité des étoiles durant la période $[T_1; T_2]$ basé sur un parcours en largeur inversé de l'arbre. Ce type de configuration est appelé cycle global montant parce qu'il favorise le flux montant du trafic prioritaire : une trame émise en trafic prioritaire par une feuille peut être remontée vers le haut de l'arbre (par exemple le CPAN) en un cycle global (sauf cas de saturation du réseau pour ce type de trafic). Cette configuration est utilisée pour les applications de collecte. Inversement, le cycle global descendant repose sur une organisation de l'activité des étoiles basée sur un parcours en largeur de l'arbre : le trafic prioritaire descendant est favorisé permettant à une trame émise par le CPAN d'atteindre l'étoile de sa destination en un cycle global.

MaCARI peut alterner cycle global montant et cycle global descendant pour rendre les

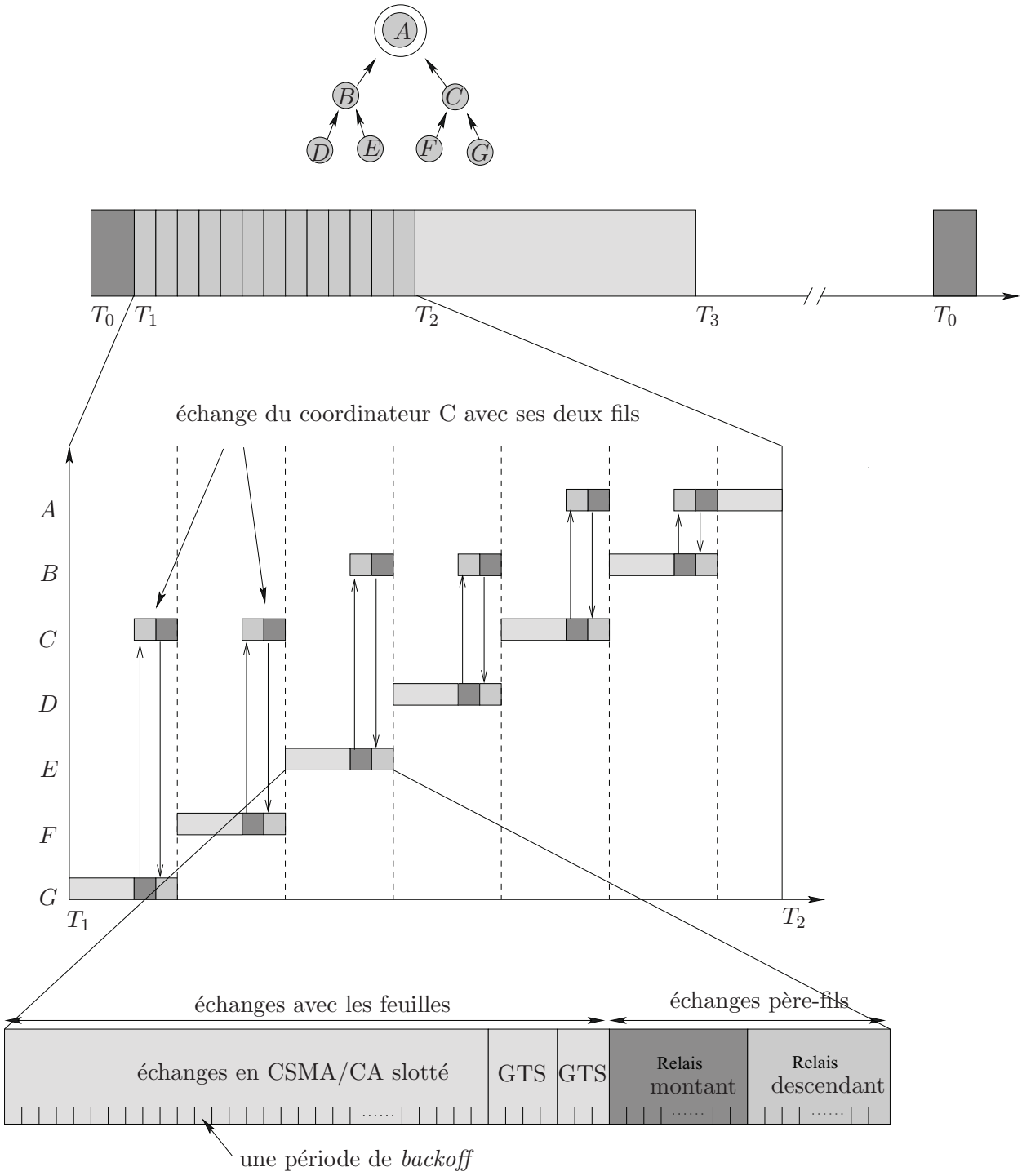


Figure 2.15 – Décomposition d'un cycle MaCARI.

flux de trafic homogènes dans le cas d'applications impliquant des échanges à destination quelconque. L'état montant ou descendant d'un cycle est indiqué dans le *beacon* (codé sur un bit). L'algorithme de calcul du séquencement des périodes d'activité des étoiles pour $[T_1; T_2]$ est simple et facile à réaliser : il repose sur le parcours en largeur ou le parcours en largeur inversé. Cette alternance permet de borner l'acheminement d'une trame non-reportée (émise lors du cycle de sa génération) de bout en bout. On peut alors borner dans le temps le délai d'acheminement d'une trame selon la durée d'un cycle. Or, le pire des cas correspond à une trame générée par une feuille à la fin de sa période de GTS lorsque la durée restante dans cette période n'est pas suffisante pour envoyer la trame : l'envoi de la trame est donc reporté dans le cycle suivant.

La figure 2.16 illustre la configuration constituant le pire des cas pour lequel l'acheminement d'une trame est étalé sur cinq cycles MaCARI lorsqu'une trame est reportée durant un cycle favorisant le flux montant. Pour chaque cycle, la figure 2.16 propose deux types de schématisation. Sur la gauche, un chronogramme indique le séquencement des périodes d'activité (en gris clair) et les relais père fils (en gris foncé). Sur la droite, une représentation de l'arbre permet de suivre les sauts et l'évolution de la position de la trame au cours de ce cycle. La figure 2.16 illustre le cas pour lequel la feuille f_A émet une trame qui a pour destination la feuille f_X, soit un parcours de l'arbre d'une extrémité à l'autre.

Le cycle 1 favorise le trafic descendant, la trame est générée par la feuille f_A en fin de période de GTS et la durée restante de ce GTS ne permet pas que la trame soit envoyée : elle subit donc un report d'envoi au cycle suivant. Le cycle 2 favorise le trafic prioritaire descendant : la feuille f_A émet la trame à son coordinateur A (saut 1) qui la transmet à son tour à son coordinateur père B (saut 2). Ensuite, lors du cycle 3, favorisant le flux prioritaire montant, la remontée de la trame de B au CPAN est symbolisée par une flèche en pointillé : elle représente la suite de sauts réalisés lors des intervalles de relais entre les coordinateurs intermédiaires liant B et le CPAN. Puis, pendant le quatrième cycle qui favorise à nouveau le trafic prioritaire descendant, la trame est acheminée au coordinateur X de la même manière que dans le cycle précèdent. Enfin, lors du cinquième cycle, le coordinateur X transmet la trame à sa feuille f_X durant la période d'activité intra-étoile.

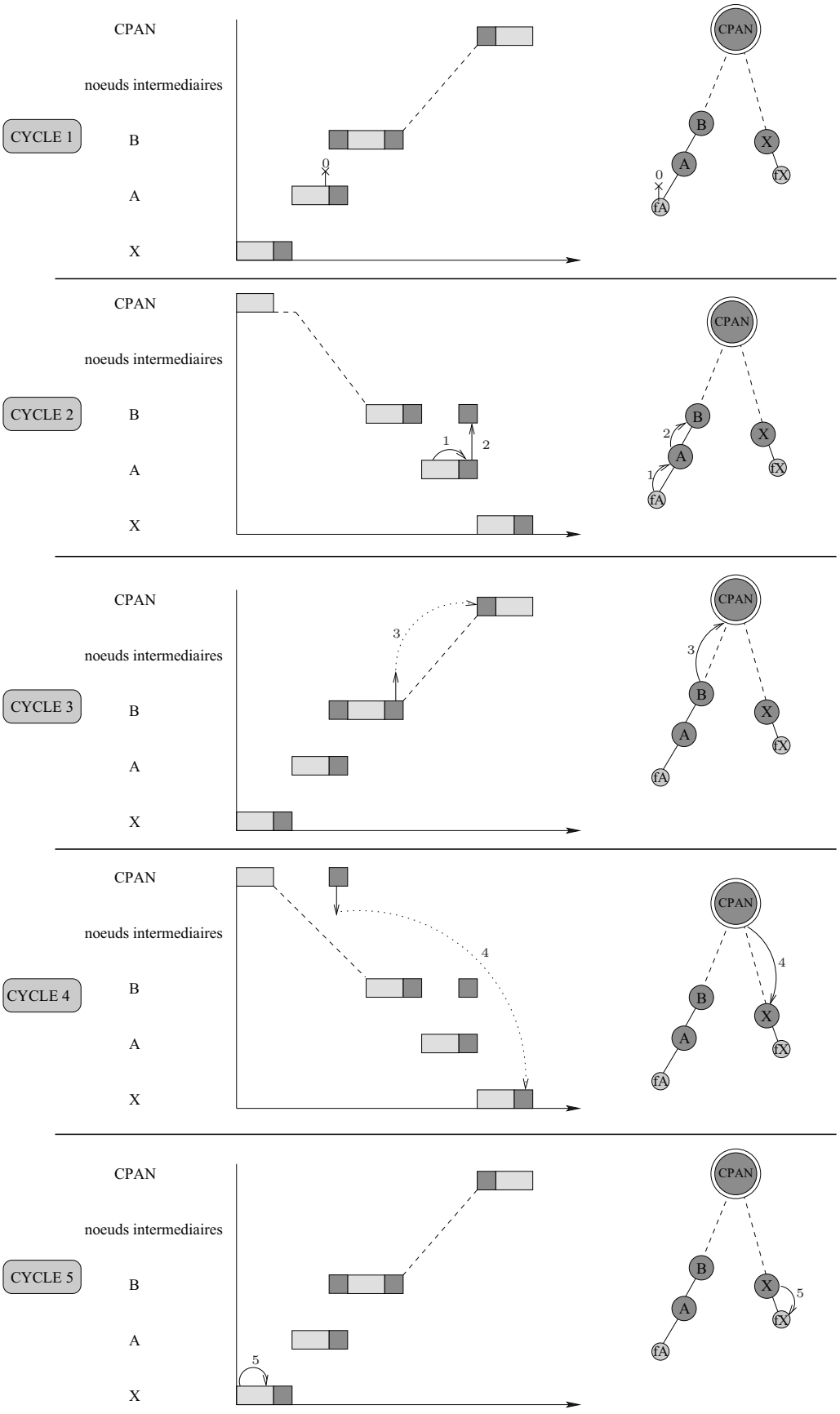


Figure 2.16 – Représentation du délai d'acheminement maximum de bout en bout dans MaCARI.

Ce pire des cas permet d'identifier le délai d'acheminement de bout en bout maximal, c'est-à-dire la durée entre l'émission et la réception d'une trame par le destinataire, selon la destination du trafic lorsque le séquencement des périodes d'activité est alterné.

Si la destination est un coordinateur (cas illustré du cycle 1 au cycle 4 sur la figure 2.16), nous comptons un cycle durant lequel la trame à émettre au début de $[T_1; T_2]$ est reportée, un cycle durant lequel la trame est acheminée au coordinateur de l'étoile de niveau $n-1$, un cycle pour que la trame soit remontée au CPAN, et une durée $[T_1; T_2]$ pour atteindre le coordinateur cible : le délai est donc inférieur à 3 cycles + la durée de l'intervalle de temps $[T_1; T_2]$.

Si la destination est une feuille (cas illustré du cycle 1 au cycle 5 sur la figure 2.16), nous comptons un cycle durant lequel la trame à émettre au début de $[T_1; T_2]$ est reportée, un cycle durant lequel la trame est acheminée au coordinateur de l'étoile de niveau $n-1$, un cycle pour que la trame soit remontée au CPAN et un cycle jusqu'à la prochaine période d'activité de l'étoile : le délai est donc inférieur à 4 cycles.

2.2.9 Conclusion

Le protocole MaCARI est un protocole MAC économe en énergie qui fournit une différenciation de service selon le type de trafic. L'économie d'énergie se réalise *via* la mise en veille des entités du réseau pendant certaines périodes. La différenciation de service est elle réalisée en utilisant deux mécanismes de communication. Les entités d'un réseau MaCARI sont synchronisées entre elles ce qui permet d'organiser un calendrier définissant des périodes d'activité et d'inactivité.

2.3 Plateforme de développement

La plateforme de développement choisie par l'équipe Réseaux et Protocoles est composée de cartes Crossbow TelosB [25], dont une photo apparait dans la figure 2.17. C'est une plateforme basée sur un sytème *open-source* (libre accès au code source et à sa redistribution) basse consommation destinée au domaine des réseaux de capteurs. Ce support vise à s'affranchir des contraintes liées à la solution propriétaire Télit utilisée dans l'implémentation matérielle existante de MaCARI.

TelosB est un type de carte très répandu dans le domaine de la recherche impliquant les réseaux de capteurs sans fils. Il a été développé par l'université de Berkley et repose sur le système d'exploitation TinyOS [26] programmé à partir du langage NesC [27].

2.3.1 Plateforme matérielle Crossbow TelosB



Figure 2.17 – Carte Crosbow TelosB.

Les principales caractéristiques d'une carte TelosB sont les suivantes :

- Un module radio CC2420 [28] (conçu pour la norme IEEE 802.15.4) et une antenne interne. La plage de fréquence exploitée est de 2.4 à 2.4835 GHz. Le débit théorique est de 250 kbps et la portée maximale en ligne de vue et en milieu ouvert est d'environ 150 mètres, avec une puissance de transmission de 0 dBm (mesures issues de [29]).
- Un microprocesseur 8 bits MSP430 cadencé à 8 MHz. Il dispose de 10 Ko de RAM ainsi que de 48 ko de mémoire flash.
- Un emplacement pouvant accueillir 2 piles de type AA.
- Des capteurs (en option) de luminosité, d'humidité et de température.
- Une interface USB pour la programmation.

2.3.2 Le Système d'exploitation TinyOS et le langage NesC

TinyOS est un système d'exploitation *open-source* dédié aux réseaux de capteurs sans fil. Son objectif est de fournir un système d'exploitation de taille mémoire réduite pour répondre aux contraintes d'espace mémoire liés aux réseaux de capteurs. TinyOS est

développé, suivi et soutenu par l'université américaine de Berkeley située en Californie. Il est disponible en téléchargement sous la licence BSD *(Berkeley Software Distribution)*. TinyOS est programmable dans un langage évènementiel orienté composants appelé NesC. NesC, pour *network embedded system C*, est un langage de programmation issu du *C* basé sur l'utilisation de composants (représentant des éléments matériels ou logiciels). Une application est constituée d'un ensemble de composants reliés entre eux par des interfaces.

TinyOS repose sur une gestion évènementielle : l'activation de tâches et leur interruption s'effectuent à l'apparition d'évènements. Ce type de fonctionnement a pour avantage d'être économe en énergie : lorsqu'aucune tâche n'est active, le système peut passer en état de veille. D'autre part, TinyOS n'est pas préemptif au niveau des tâches mais les interruptions matérielles sont prioritaires. Ainsi, les tâches entre-elles ne s'interrompent pas mais une interruption peut stopper l'exécution d'une tâche. TinyOS dispose donc de deux niveaux de priorité (synchrone pour les tâches et asynchrone pour les interruptions) : cela limite les caractéristiques temps réel du système, mais cela permet une simplification importante de la programmation et évite la surcharge liée à la gestion de tâches concurrentes.

TinyOS est un système ouvert très utilisé pour la programmation de capteurs sans fil. L'approche modulaire des composants permet de faciliter la programmation d'applications sur différentes plateformes matérielles. Le développement basé sur TinyOS est très actif ; il existe de nombreuses implémentations de protocoles récents [30].

Chapitre 3

Contribution : implémentation de MaCARI sur TelosB

L'objectif de ce stage est la réalisation d'une implémentation fonctionnelle du protocole MaCARI sur TelosB et la mise en œuvre d'une application dans une mine industrielle. MaCARI étant basé sur le standard IEEE 802.15.4, le développement de MaCARI sur TelosB repose sur l'une des implémentations existantes du standard.

Tout d'abord, nous décrivons l'analyse menant au choix de la base logicielle de l'implémentation. Puis, nous présentons l'architecture de l'implémentation et ses fonctionnalités. Enfin, nous décrivons les principales optimisations réalisées au cours du développement.

3.1 Base logicielle de l'implémentation de MaCARI

Il existe deux implémentations *open-source* du standard IEEE 802.15.4 fonctionnelles pour la plateforme TelosB. Tout d'abord, nous présentons l'implémentation de l'Institut Polytechnique de Porto. Ensuite, nous décrivons l'implémentation de l'Université Technique de Berlin. Enfin, nous présentons un bilan expliquant le choix que nous avons fait pour la base logicielle de l'implémentation de MaCARI.

3.1.1 Implémentation du standard IEEE 802.15.4 de l'Institut Polytechnique de Porto

Le CISTER *(Research Centre in Real-Time Computing Systems)* est un laboratoire de recherche basé à l'Institut Polytechnique de Porto au Portugal. Au sein de ce laboratoire, l'équipe IPP-HURRAY (Institut Polytechnique de Porto - *HUgging Real-time and Reliable Architectures for computing sYstems*) a réalisé en 2007 une implémentation [31] du standard IEEE 802.15.4 pour la plateforme TelosB. Nous nommerons cette implémentation CISTER15.4 dans la suite de ce mémoire.

Tout d'abord, nous décrivons les fonctionnalités fournies par cette implémentation. Puis, nous présentons son architecture. Enfin, nous détaillons ses principales caractéristiques.

3.1.1.1 Fonctionnalités supportées par CISTER15.4

La version actuelle de l'implémentation CISTER15.4 supporte les fonctionnalités du standard IEEE 802.15.4 suivantes :

- l'algorithme de CSMA/CA slotté,
- les mécanismes liés aux GTS,
- la gestion de la transmission des données de manière directe, indirecte et par GTS,
- la gestion du *beacon* et de la synchronisation,
- l'encapsulation des trames,
- les mécanismes d'association et de désassociation,
- la gestion de la PIB *(Pan Information Based)* de la couche MAC,
- le *scan* de canal passif et le *scan* de détection d'énergie.

En revanche, les fonctionnalités suivantes ne sont pas disponibles :

- l'algorithme de CSMA/CA non slotté,
- le *scan* de canal actif et le *scan* d'orphelin,
- la gestion de la sécurité.

3.1.1.2 Architecture de CISTER15.4

L'architecture de l'implémentation CISTER15.4 est basée sur l'abstraction de chacune des couches du standard IEEE 802.15.4. Chaque couche est contenue dans un composant

logiciel fournissant des interfaces et les fonctionnalités correspondantes à son rôle dans la pile protocolaire [31].

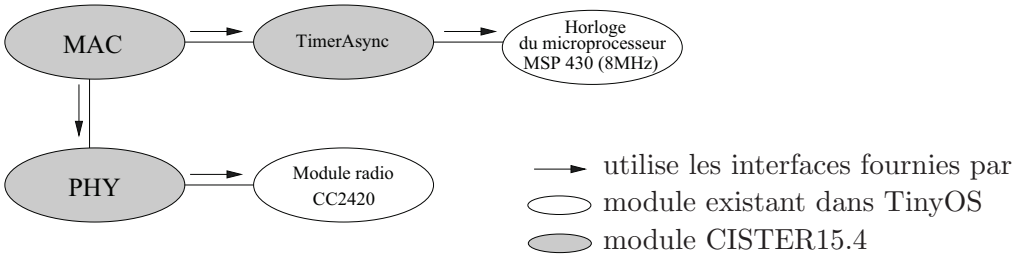


Figure 3.1 – Organisation de l'architecture de l'implémentation CISTER15.4.

La figure 3.1 décrit la structure de l'implémentation. Le module radio CC2420 est géré par le composant logiciel représentant la couche PHY. Au niveau MAC s'effectue la gestion du temps (déterminante pour la synchronisation), le découpage des cycles et les mécanismes de transmission de trames. *Timerasync* est un *timer* (compteur basé sur l'horloge du microprocesseur) : il est utilisé pour déclencher des interruptions liées à un évènement.

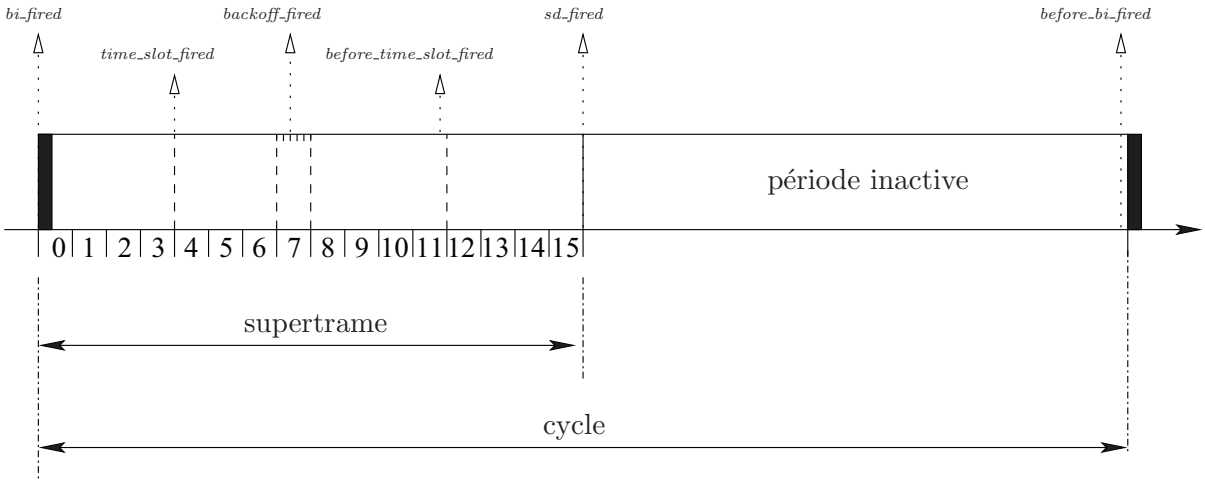


Figure 3.2 – Représentation des évènements du *timer*.

La figure 3.2 présente les différents évènements du *timer*. L'évènement *bi_fired* signale le déclenchement d'un nouvel intervalle de *beacon*. L'évènement *before_bi_fired* est signalé juste avant la fin d'un intervalle de *beacon* (la durée entre *before_bi_fired* et *bi_fired* est une constante paramétrable). *time_slot_fired* signale le début d'un slot de temps (utilisé par les mécanismes liés au GTS notamment). *before_time_slot_fired* est un évènement signalé

juste avant le début d'un slot de temps (cette durée est paramétrable) ; il sert à situer l'instant de réveil des nœuds afin qu'ils aient le temps de réaliser les traitements préalables au début du prochain cycle. L'évènement *backoff_fired* signale chaque début de période de *backoff* (utilisé par le CSMA/CA slotté ou non). Enfin, *sd_fired* désigne la fin de la supertrame et le début de la période d'inactivité.

3.1.1.3 Caractéristiques de CISTER15.4

Les caractéristiques présentées sont liées à la documentation disponible, à la lisibilité du code, ainsi qu'aux évaluations de l'implémentation CISTER15.4.

Documentation : L'implémentation CISTER15.4 est bien documentée. Le guide technique [31] est détaillé et précis. Il fournit la description de chaque composant logiciel ainsi que la description de chacune des fonctions et variables utilisées.

Lisibilité du code et modularité : L'abstraction consistant à associer une couche à un composant logiciel NesC permet d'identifier naturellement les fonctionnalités du standard IEEE 802.15.4. Le code est réparti sur une faible quantité de fichiers de travail : il est par conséquent moins modulaire mais le développement est simplifié.

Évaluation : Les évaluations disponibles ont été réalisées par le groupe CISTER dans le cadre de sujets de thèses [32], [33]. Hormis une évaluation du CSMA/CA slotté pour une topologie en étoile [32], ces évaluations visent plutôt à valider les fonctionnalités disponibles sans réels tests de performances (synchronisation, échanges simples de trames, association et désassociation, et acquittements).

3.1.2 Implémentation du standard IEEE 802.15.4 de l'Université Technique de Berlin

Le TKN *(Telekommunikationsnetze)* est un département de recherche basé à l'Université Technique de Berlin en Allemagne. J'ai été en contact avec J. H. Hauer, ingénieur de recherche au sein de ce département, qui participe activement au développement du

système TinyOS sur lequel il a réalisé une implémentation du standard IEEE 802.15.4 appelée TKN15.4 [34].

Comme pour CISTER15.4, nous décrivons d'abord les fonctionnalités fournies par cette implémentation. Puis, nous présentons son architecture et son organisation. Enfin nous détaillons ses principales caractéristiques.

3.1.2.1 Fonctionnalités supportées par TKN15.4

La version actuelle de l'implémentation TKN15.4 supporte les fonctionnalités du standard IEEE 802.15.4 suivantes :

- l'algorithme de CSMA/CA slotté et non slotté,
- la gestion du *beacon* et de la synchronisation,
- l'encapsulation des trames,
- la gestion de la transmission des données de manière directe et indirecte,
- les mécanismes d'association et de désassociation,
- la gestion de la PIB de la couche MAC,
- les quatre types de *scan* de canal.

En revanche, les fonctionnalités suivantes ne sont pas supportées :

- la CFP et les mécanismes liés aux GTS,
- la gestion de la sécurité.

3.1.2.2 Architecture de TKN15.4

L'architecture de l'implémentation TKN15.4 est organisée pour répondre à trois objectifs au niveau MAC : l'indépendance vis à vis de la plateforme matérielle, la modularité et l'évolutivité. L'indépendance matérielle favorise la portabilité de l'implémentation. La modularité du code permet l'exclusion de composants logiciels inutiles afin de réduire la taille du code et d'économiser l'espace mémoire disponible sur les cartes. L'évolutivité simplifie les ajouts et les modifications de fonctionnalités.

La figure 3.3 [34] présente l'organisation des composants logiciels de la couche MAC de TKN15.4. Les composants sont reliés entre eux par des interfaces représentées par les flèches. Dans cette figure, la couche PHY et l'horloge matérielle apportent les interfaces nécessaire à la couche MAC. La couche MAC est composée d'interfaces et de trois

niveaux de composants fournissant des fonctionnalités décrites dans le tableau 3.1. Les composants de bas niveau (boîtes en gris foncé) organisent l'accès à la ressource radio. La plupart des composants du niveau intermédiaire (boîtes en gris clair) sont responsables de la réalisation des différentes étapes d'un cycle. Les composants de haut niveau (boîtes blanches) implémentent les services restants de contrôle et de données qui fournissent les primitives MLME et MLDE à la couche supérieure.

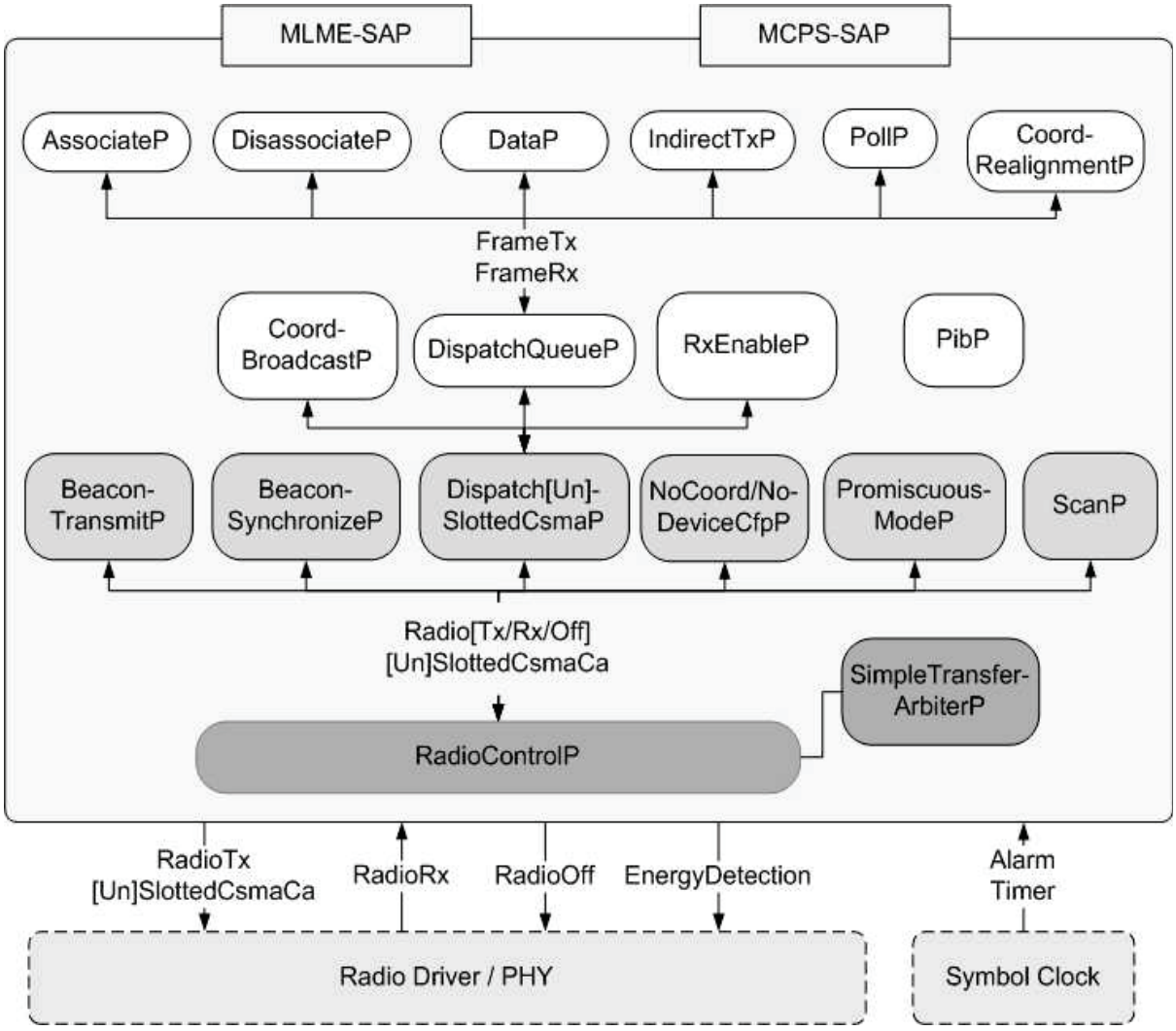


Figure 3.3 – Représentation des composants logiciels de la couche MAC TKN15.4.

La ressource radio est gérée par un système de jeton afin d'éviter les conflits d'accès. Avant qu'un composant puisse accéder à la ressource radio, il doit faire une requête à un composant arbitre jouant le rôle de sémaphore. Une fois que l'accès est autorisé par l'arbitre, le composant peut utiliser la ressource de manière exclusive. Après utilisation, la

Tableau 3.1 – Description des fonctions fournies par les composants logiciels de TKN15.4.

Composants / Interfaces	**Fonctions [primitives du standard IEEE 802.15.4 associées]**
Alarm / Timer	Interfaces liées à la gestion du temps (alarme et compteur de temps)
AssociateP	Association [MLME-ASSOCIATE,MLME-COMM-STATUS]
BeaconTransmitP	Transmission périodique du *beacon* [MLME-START]
BeaconSynchronizeP	Synchronisation périodique avec le *beacon* [MLME-SYNC(-LOSS), MLME-BEACON-NOTIFY]
CoordBroadcastP	Transmission des trames *broadcast* du coordinateur
CoordRealignmentP	Gestion des commandes de resynchronisation [MLME-ORPHAN, MLME-COMM-STATUS]
DataP	Création et purge des trames de données [MCPS-DATA, MCPS-PURGE]
DisassociateP	Désassociation [MLME-DISASSOCIATE]
DispatchQueueP	Gestion des files d'attente pour les trames de contrôle et de données
DispatchSlottedCsmaP	Transmission/réception de trames pendant la CAP (ne contient pas l'algorithme du CSMA/CA)
DispatchUnslottedCsmaP	Transmission/réception de trames en mode non suivi de *beacon* (ne contient pas l'algorithme du CSMA/CA)
EnergyDetection	Interface permettant le *scan* d'énergie sur un canal
Frame Rx	Interface utilisée pour la réception de trame
Frame Tx	Interface commandant l'envoi d'une trame
IndirectTxP	Gestion des transmissions indirectes
NoCoordCFP	Composant vide (les GTS ne sont pas implémentés)
PibP	Gestion de la base d'information du PAN [MLME-RESET, MLME-GET, MLME-SET]
PollP	Requête de données depuis un coordinateur [MLME-POLL]
PromiscuousModeP	Activation /désactivation du mode *sniffer*
Radio Rx	Interface d'activation de la réception radio
Radio Tx	Interface d'activation de la transmission radio
Radio Off	Interface commandant la mise en sommeil du module radio
RadioClientC	Accès virtuel au composant *RadioControlP*
RadioControlImplP	Contrôle l'accès à la ressource radio
RadioControlP	Configuration du composant *RadioControlImplP*
RxEnableP	Activation de la réception radio pendant les périodes d'inactivité [MLME-RX-ENABLE]
ScanP	*Scans* de canaux [MLME-SCAN]
SimpleTransferArbiterP	Arbitrage de la ressource radio
TKN154BeaconEnabledP	Configuration de la couche MAC en mode suivi de *beacons*
TKN154NonBeaconEnabledP	Configuration de la couche MAC en mode non suivi de *beacons*

ressource doit être libérée afin que le composant suivant y ait accès. L'arbitre ordonnance le passage de jeton soit selon la méthode du tourniquet, soit selon la méthode premier arrivé premier servi. Le jeton peut être transmis d'un composant à un autre afin de pouvoir séquencer l'activité radio des composants selon les besoins.

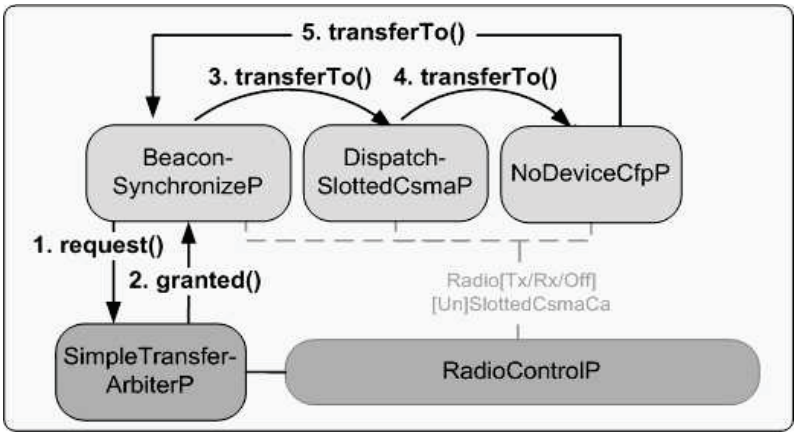


Figure 3.4 – Représentation d'un exemple de transmission du jeton entre les composants.

La figure 3.4 [34] illustre l'exemple du comportement de l'implémentation en mode suivi de beacons, pour une feuille qui doit envoyer une trame en CSMA/CA au coordinateur avec lequel elle est associée. Dans un premier temps, la couche MAC fait une demande de synchronisation avec le *beacon* du coordinateur. Le module responsable de la synchronisation, *BeaconSynchronizeP*, émet une requête *(request)* de jeton à l'arbitre radio. Lorsque que *BeaconSynchronizeP* reçoit le jeton *(request granted)*, il possède un accès exclusif à la radio qu'il place en mode réception pour attendre le *beacon*. Une fois le *beacon* reçu et traité, *BeaconSynchronizeP* transmet le jeton directement, sans passer par l'arbitre, au module responsable de la transmission de la trame en CSMA/CA slotté. Quand la CAP est terminée, le jeton est transféré au composant de gestion de la CFP (non implémenté), puis transféré au composant *BeaconSynchronizeP*, qui le rend à l'arbitre.

3.1.2.3 Caractéristiques de TKN15.4

Les caractéristiques présentées sont liées à la documentation disponible, à la lisibilité du code, ainsi qu'aux performances de l'implémentation TKN15.4.

Documentation : La documentation de TKN15.4 est incomplète. L'implémentation est récente (2009). Il existe actuellement une présentation générale [34] mais pas de guide technique à proprement dit.

Lisibilité du code et modularité : Le code est conforme au modèle de TinyOS : il est fortement modulaire. Les composants sont répartis dans de nombreux fichiers imbriqués ; la structure du code est donc plus complexe et moins naturelle à parcourir que s'il était moins disséminé.

Évaluation : À notre connaissance, aucun test de performance n'a été publié à ce jour. Néanmoins, l'architecture de l'implémentation vise particulièrement à optimiser les besoins de réactivité de la méthode d'accès au médium : l'algorithme de CSMA/CA est inclus dans la gestion du module radio afin de réduire la latence des opérations critiques de transmission (CCA, acquittements et tirage de *backoffs*). En effet, l'implémentation utilise les fonctionnalités d'envoi précédé d'un CCA automatique (utilisé après un premier CCA positif) et la gestion matérielle des acquittements fournis par le CC2420. Cette propriété permet de fournir à l'implémentation un comportement rapide et des performances élevées à ce niveau. De plus, la gestion de la ressource radio exclue la possibilité d'accès concurrent, ce qui assure une stabilité renforcée.

3.1.3 Bilan

Le choix de la base logicielle de l'implémentation de MaCARI repose sur des besoins fonctionnels, des besoins de performance et de modularité. Si les implémentations CISTER15.4 et TKN15.4 fournissent toutes deux la plupart des fonctionnalités du standard IEEE 802.15.4, elles possèdent chacune une structure propre. Le tableau 3.2 montre une comparaison des deux implémentations.

CISTER15.4 possède les mécanismes de gestion des GTS, une documentation détaillée et une plus grande ancienneté que TKN15.4. Cependant, l'architecture de TKN15.4 fournit une meilleure prise en compte des impératifs temporels liés à la méthode d'accès CSMA/CA du standard IEEE 802.15.4 sur lequel se base MaCARI. D'autre part, la gestion de la ressource radio par transmission de jeton de TKN15.4 permet une stabilité ac-

Tableau 3.2 – Bilan comparatif entre CISTER15.4 et TKN15.4.

Caractéristiques	CISTER15.4	TKN15.4
Documentation	+	-
Fonctionnalités	+	-
Modularité	-	+
Performances estimées	-	+

crue et des possibilités d'évolutions importantes : l'ajout de composants supplémentaires est simplifié. Par conséquent, nous avons choisi de baser l'implémentation de MaCARI sur TKN15.4.

3.2 Implémentation de MaCARI sur TelosB

L'implémentation de MaCARI sur TelosB a été réalisée dans l'objectif d'être utilisée par une application mise en œuvre dans un contexte industriel minier. Pour cela, certains paramètres ont été fixés (rendus statiques) et un sous-ensemble des fonctionnalités a été mis de côté. Ces dispositions ont été prises pour que la durée du développement corresponde au calendrier du stage et qu'une version soit exploitable pour la phase de mise en œuvre au Canada au sein du laboratoire LRTCS de Val-d'Or.

Durant le développement de MaCARI sur la base TKN15.4, j'ai bénéficié des conseils de J.H. Hauer, l'auteur de TKN15.4. Il m'a autorisé à publier certaines parties techniques des correspondances (en anglais) que nous avons échangées par messagerie électronique (jointes en annexe A).

Tout d'abord, nous présentons les fonctionnalités retenues pour l'implémentation de MaCARI. Ensuite, nous décrivons l'architecture et les mécanismes de l'implémentation. Enfin, nous nous concentrons sur les choix d'optimisation réalisés dans la phase de développement.

3.2.1 Fonctionnalités

L'implémentation de MaCARI sur TelosB fournit les fonctionnalités suivantes :

– l'algorithme de CSMA/CA slotté,

– la gestion des intervalles de relais,

– la gestion de la synchronisation dans l'arbre (par cascade de *beacons*),
– la gestion dynamique du découpage temporel du cycle MaCARI,
– la création statique de la topologie du réseau,
– la différenciation de trafic,
– l'adressage hiérarchique et le routage hiérarchique,
– l'estampillage temporel des trames.

Certaines des fonctionnalités ont été mises de côté ou rendues statiques. Elles n'ont pas d'impact profond sur les principes de la couche MaCARI. Les fonctionnalités qui ne sont pas supportées dans cette version de l'implémentation sont les suivantes :

– la gestion dynamique de la topologie du réseau (mécanismes d'association et de désassociation),
– la gestion des GTS durant les périodes d'activité intra-étoile,
– l'optimisation de l'économie d'énergie durant la période $[T_0; T_1]$ présentée dans [8],
– la gestion des files d'attente pour les trames de données dans une topologie avec plus de deux coordinateurs fils par coordinateur ($Rm > 2$).

3.2.2 Architecture et mécanismes de l'implémentation

L'architecture est basée sur l'implémentation TKN15.4 : elle reprend la couche PHY de TKN15.4 et conserve une architecture modulaire (on retrouve la gestion de la ressource radio par un mécanisme de jeton). Néanmoins, l'implémentation de MaCARI nécessite une phase de développement complexe impliquant l'entière refonte des principaux composants au niveau MAC. La figure 3.5 représente le diagramme de l'organisation des composants logiciels dans l'implémentation de MaCARI. Nous décrivons par la suite le rôle de chacun des composants et des interfaces des couches PHY et MaCARI.

Il existe un profil de code pour chacun des trois rôles qu'une entité peut avoir dans MaCARI (CPAN, coordinateur, feuille) : l'architecture du code permet d'optimiser l'espace mémoire utilisé en ôtant les composants et les fonctionnalités inutiles selon le rôle de l'entité dans l'arbre.

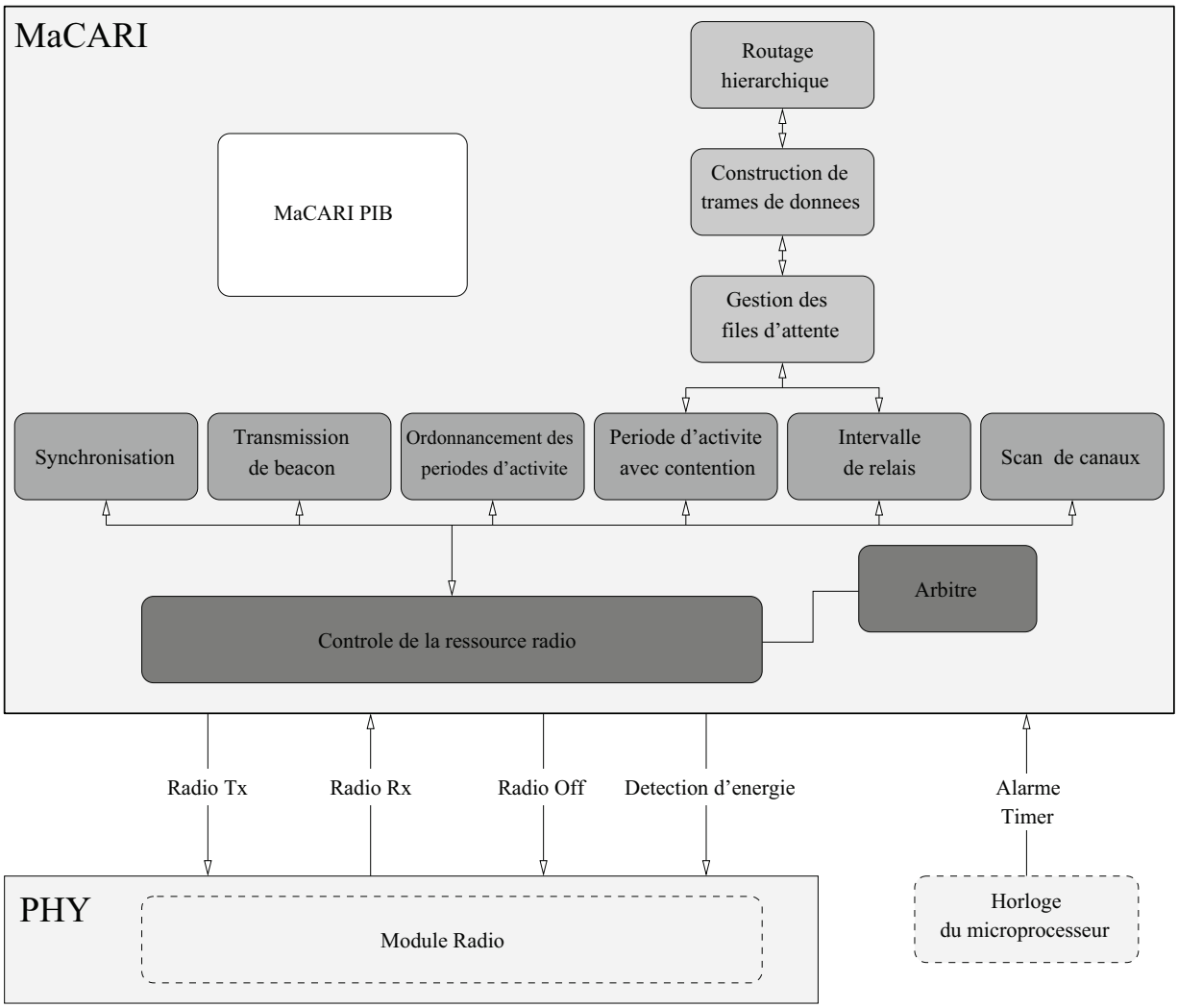


Figure 3.5 – Représentation de l'architecture de l'implémentation de MaCARI.

3.2.2.1 Les composants et les interfaces de la couche PHY

La couche PHY pilote le module radio dont l'accès est possible par les interfaces suivantes :

- *Radio Tx*, qui commande la transmission radio et place les trames dans la mémoire tampon *(buffer)* d'envoi du module radio,
- *Radio Rx*, qui permet de placer le module radio en mode réception et qui indique à la couche MaCARI la réception d'une trame par le biais d'une interruption,
- *Radio Off* , qui permet de mettre le module radio en sommeil afin d'économiser de l'énergie,
- *Détection d'énergie*, qui fournit à la couche MaCARI l'interface liée aux *scans* et qui permet d'obtenir l'énergie maximale détectée sur un canal donné pendant une durée paramétrable.

Pour la transmission des trames, l'intégration du CSMA/CA dans la couche PHY de TKN15.4 (voir 3.1.2.3) est reprise dans cette implémentation : la couche MaCARI spécifie les paramètres initiaux de CSMA/CA mais l'algorithme en lui même est implémenté et exécuté au sein du pilote du module radio. Les acquittements sont gérés matériellement au niveau de la couche PHY [28].

3.2.2.2 Les composants et les interfaces de la couche MaCARI

La couche MaCARI a une référence de temps basée sur l'horloge du microprocesseur MSP430 qui, par le biais d'interfaces, fournit des fonctions de *timer* et d'alarme. Comme l'illustre la figure 3.5, la couche MaCARI est composée de quatre familles de composants : les composants organisant l'accès à la ressource radio (en gris foncé), les composants qui ont un besoin d'accès direct à la radio (en gris intermédiaire), les composants de routage, de construction, et de stockage des trames de données (en gris pâle) et enfin, le composant de configuration (en blanc).

Les composants qui organisent l'accès à la radio

Ces composants ont le même mode de fonctionnement que dans TKN15.4 : le composant arbitre autorise l'accès au composant de contrôle de la ressource radio en attribuant

un jeton à un composant client qui lui en a fait la requête ; par la suite, un composant qui possède le jeton peut soit rendre le jeton à l'arbitre soit le transmettre à un autre composant. L'arbitre dans cette implémentation organise la distribution par un ordonnancement en tourniquet *(round robin)*.

Les composants qui ont un besoin d'un accès direct à la ressource radio

Ces composants sont appelés des composants clients radio. Il existe deux types de composants clients radio : les clients indirects qui ne peuvent recevoir le jeton que par l'intermédiaire d'autres clients (transfert de jeton) et les clients directs qui peuvent en plus recevoir le jeton de l'arbitre (suite à l'envoi d'une requête de jeton). C'est par ce mécanisme de requête et de transfert de jeton que l'activité radio est séquencée dans cette implémentation de MaCARI.

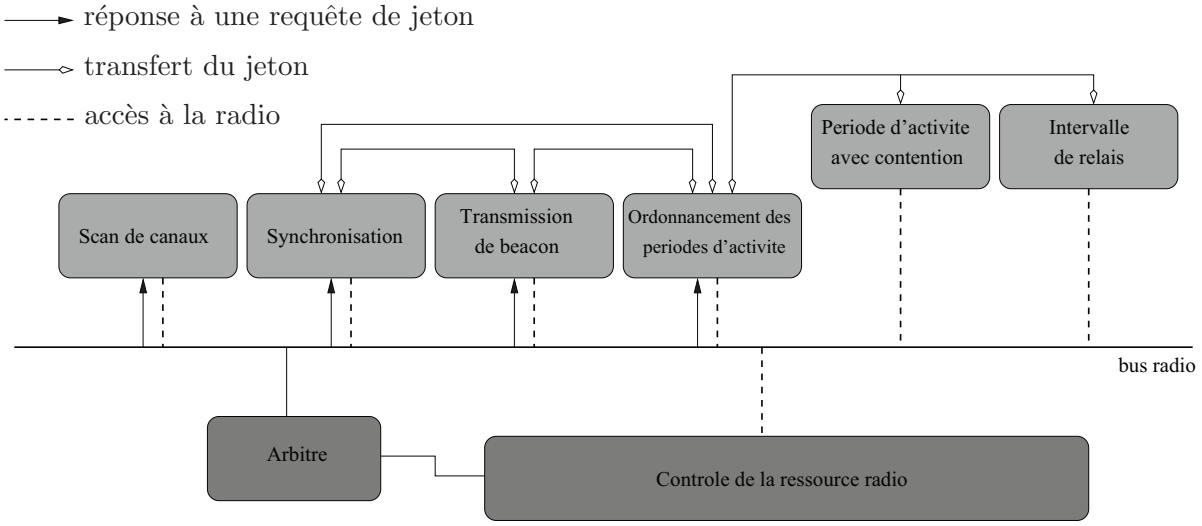


Figure 3.6 – Représentation des mécanismes de passage de jeton et de l'accès à la ressource radio.

La figure 3.6 représente la gestion de la ressource radio par le mécanisme de jeton. Les composants *Scan de canaux, Synchronisation, Transmission de beacon* et *Ordonnancement des périodes d'activité* sont des clients directs. Les composants *Période d'activité avec contention* et *Intervalle de relais* sont des clients indirects. Le client qui possède le jeton a un accès exclusif à la ressource radio. Le jeton peut être transféré d'un composant à un autre. Ces possibles transferts de jetons sont représentés sur la figure par les liens avec des flèches blanches.

Le composant *Scan de canaux* : Il remplit les fonctionnalités similaires au composant *ScanP* de TKN15.4 : il permet de réaliser les quatre types de *scans* de canaux de la norme IEEE 802.15.4. Comme la création et la gestion de la topologie sont statiques, seuls les *scans* d'énergie et les *scans* passifs sont utilisés pour cette version de l'implémentation de MaCARI (même si les quatre types possibles sont implémentés).

Le composant *Transmission de beacon* : Ce module est utilisé par le CPAN et les coordinateurs pour transmettre leur *beacon* à un moment précis durant la période de synchronisation $[T_0; T_1]$. L'adresse de l'entité qui transmet un *beacon* est indiquée dans l'entête du *beacon*, la charge utile *(payload)* contient les informations nécessaires à l'organisation du cycle MaCARI.

Pour le CPAN, le *beacon* est construit de façon statique (dans cette version, son contenu est établi lors du flashage de la carte) : il contient le nombre total de coordinateurs dans le PAN, la durée du cycle global, les paramètres de durée d'une période d'activité et la liste ordonnée des adresses des coordinateurs selon leur position dans l'arbre. Cette liste est complétée de manière statique et ordonnée selon un parcours en largeur de l'arbre dans cette version de MaCARI. Elle définit l'ordre dans lequel les coordinateurs envoient leur *beacon* et le séquencement des périodes d'activité pour la période $[T_1; T_2]$. Ce séquencement peut être inversé si le bit d'inversion vaut 1. Le CPAN envoie le *beacon* périodiquement selon la durée du cycle MaCARI (durée qui peut varier à chaque cycle).

Un coordinateur attend la réception d'un *beacon* pour se synchroniser et déterminer sa position dans l'arbre afin d'en déduire à quel moment il doit transmettre son propre *beacon*. Lorsque le coordinateur reçoit un *beacon* valide, ce *beacon* est traité par le composant *Synchronisation* qui crée d'abord l'entête d'un nouveau *beacon* (qui contient son adresse) puis met à jour la charge utile en fonction du *beacon* reçu. Enfin, lorsque l'instant d'envoi calculé est arrivé, le coordinateur transmet son *beacon*.

Le composant *Synchronisation* : Ce composant permet à toutes les entités de se synchroniser avec le PAN. Avant chaque début d'un cycle MaCARI, les coordinateurs et les feuilles du PAN placent leur module radio en réception. Lorsqu'une entité reçoit un *beacon*, l'instant de réception est utilisé pour déterminer l'instant T_0 du cycle Ma-

CARI (référence de temps unique dans le PAN). Ensuite, l'entité traite les informations contenues dans le *beacon*. Ainsi, selon la position de la source du *beacon* reçu, le nœud peut situer sa position dans l'arbre, celle de son coordinateur père, et, si le nœud est un coordinateur, la position de ses coordinateurs fils s'il en possède.

Enfin, le composant arme un *timer* expirant juste avant le début du prochain T_0, correspondant au début du prochain cycle MaCARI, pour préparer la prochaine réception de *beacon* : lorsque l'évènement est signalé, le module radio est placé en mode réception.

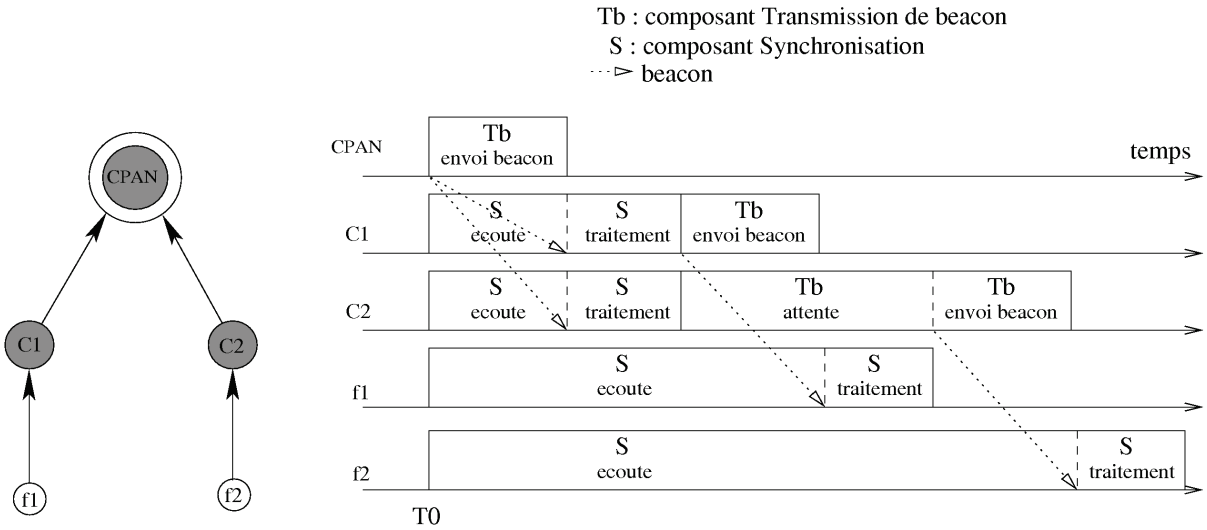


Figure 3.7 – Représentation du rôle des composants durant la période de synchronisation.

La figure 3.7 représente le rôle des composants *Synchronisation* et *Transmission de beacon* durant la période de synchronisation pour un exemple de topologie de réseau avec un CPAN et ses deux coordinateurs fils (C_1 et C_2) ayant chacun une feuille associée (respectivement f_1 et f_2). L'ordre de synchronisation correspond à un parcours en largueur de l'arbre (CPAN, C_1, C_2). À T_0, le composant *Transmission de beacon* du CPAN envoie le *beacon*. C_1 et C_2, par le biais du composant *Synchronisation*, reçoivent le *beacon* et le traitent. Une fois le traitement des informations réalisé, le jeton est transmis au composant *Transmission de beacon* de C_1 et C_2. L'instant d'envoi du *beacon* est alors calculé selon la position du nœud : C_1 envoie son *beacon*, C_2 attend son instant d'envoi. En parallèle, chacune des feuilles f_1 et f_2 utilise le composant *Synchronisation* pour recevoir et traiter le *beacon* de leur coordinateur père.

Le composant *Période d'activité avec contention* : Ce composant est responsable des périodes d'activité dans lesquelles les trames sont transmises selon l'algorithme de CSMA/CA slotté. La période est découpée en slots. La durée des slots et le nombre de slots pour une période d'activité sont indiqués dans la charge utile du *beacon*.

Il existe deux types de périodes d'activité avec contention dans MaCARI. La période intra-étoile définit la période d'échanges entre un coordinateur et ses feuilles. La période de routage est définie par l'intervalle $[T_2; T_3]$ dans laquelle seuls les coordinateurs sont actifs. Le composant est générique : il est instancié selon la période d'activité.

En début de période, le composant teste si la file d'attente correspondant à la période contient des trames en attente de transmission. Si c'est le cas, il tente de les envoyer selon l'algorithme CSMA/CA slotté. Sinon, le module radio est placé en réception (le composant conserve le jeton). Le composant teste l'état de la file d'attente en continu durant la période d'activité : si la file n'est pas vide et qu'il reste suffisament de temps avant la fin de la période, le composant tente d'envoyer les trames contenues dans la file. A la fin de la période, le jeton est rendu au composant d'*Ordonnancement des périodes d'activité*.

Le composant *Intervalle de relais* : Ce composant est responsable des échanges de trames de type prioritaire entre coordinateurs père et fils. La durée de cette période est exprimée en période de *backoff* dans le *beacon*. Elle est divisée en deux parties égales : durant la première partie, le coordinateur fils transmet les trames à son père et durant la deuxième partie, le père transmet les trames à son fils. A la fin de la période, le jeton est rendu au composant d'*Ordonnancement des périodes d'activité*.

Il existe deux instanciations possibles pour ce composant : une pour l'intervalle de relais père-fils et une pour l'intervalle de relais fils-père. Par exemple, lors d'un cycle MaCARI, un coordinateur ayant deux fils coordinateurs aura un intervalle relais fils-père avec son coordinateur père, et deux intervalles de relais père-fils avec ses coordinateurs fils.

Le composant *Ordonnancement des périodes d'activité* : Ce composant est responsable du séquencement des périodes d'activité et de la gestion correspondante du

module radio. Le jeton est transmis à ce composant une fois que les tâches de la période de synchronisation nécessitant la ressource radio sont terminées. Dans un premier temps, l'ordonnanceur détermine le calendrier d'activité pour le nœud durant le cycle MaCARI. Selon le type du nœud, sa position dans l'arbre (et pour un coordinateur, le nombre et la position de ses fils), le composant séquence les différentes périodes et détermine leurs dates de début basées sur l'instant T1 du cycle. Ensuite, le composant transfère le jeton au composant correspondant à la prochaine période d'activité définie et lui fournit la date de début et la durée de la période. À chaque fin de période, le jeton est rendu au composant d'*Ordonnancement des périodes d'activité* qui le redistribue dans la foulée selon le calendrier. Une fois toutes les périodes d'activité terminées, l'ordonnanceur abandonne le jeton. La figure 3.8 montre les échanges de jeton que peut réaliser le composant *Ordonnancement des périodes d'activité* durant un cycle MaCARI.

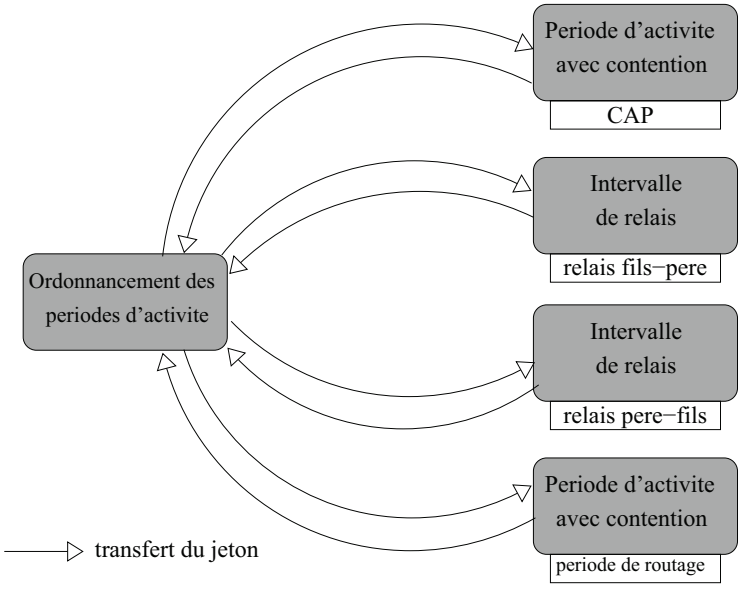


Figure 3.8 – Représentation de la gestion du jeton par l'ordonnanceur des périodes d'activité.

La figure 3.9 représente le rôle des composants *Période d'activité avec contention* (CAP) et *Intervalle de relais* (IR) durant la période $[T_1; T_3]$ pour un exemple de topologie de réseau avec un CPAN et ses deux coordinateurs fils (C_1 et C_2) ayant chacun une feuille associée (respectivement f_1 et f_2). L'ordre d'activité des nœuds correspond à un parcours en largeur inversé de l'arbre (C_2, C_1, CPAN). Le composant *Ordonnancement des périodes d'activité* détermine le calendrier, organise la distribution du jeton et instancie les

composants selon la période : intra-étoile ou routage pour le composant *Période d'activité avec contention* et relais père-fils ou relais fils-père pour le composant *Intervalle de relais*. À noter que pour tenir compte du temps nécessaire à la circulation entre les différents composants liés à l'activité, le composant *Ordonnancement des périodes d'activité* place entre chaque période un intervalle de temps appelé intervalle de garde. Ces intervalles ne sont pas représentés dans la figure 3.9 pour alléger la représentation.

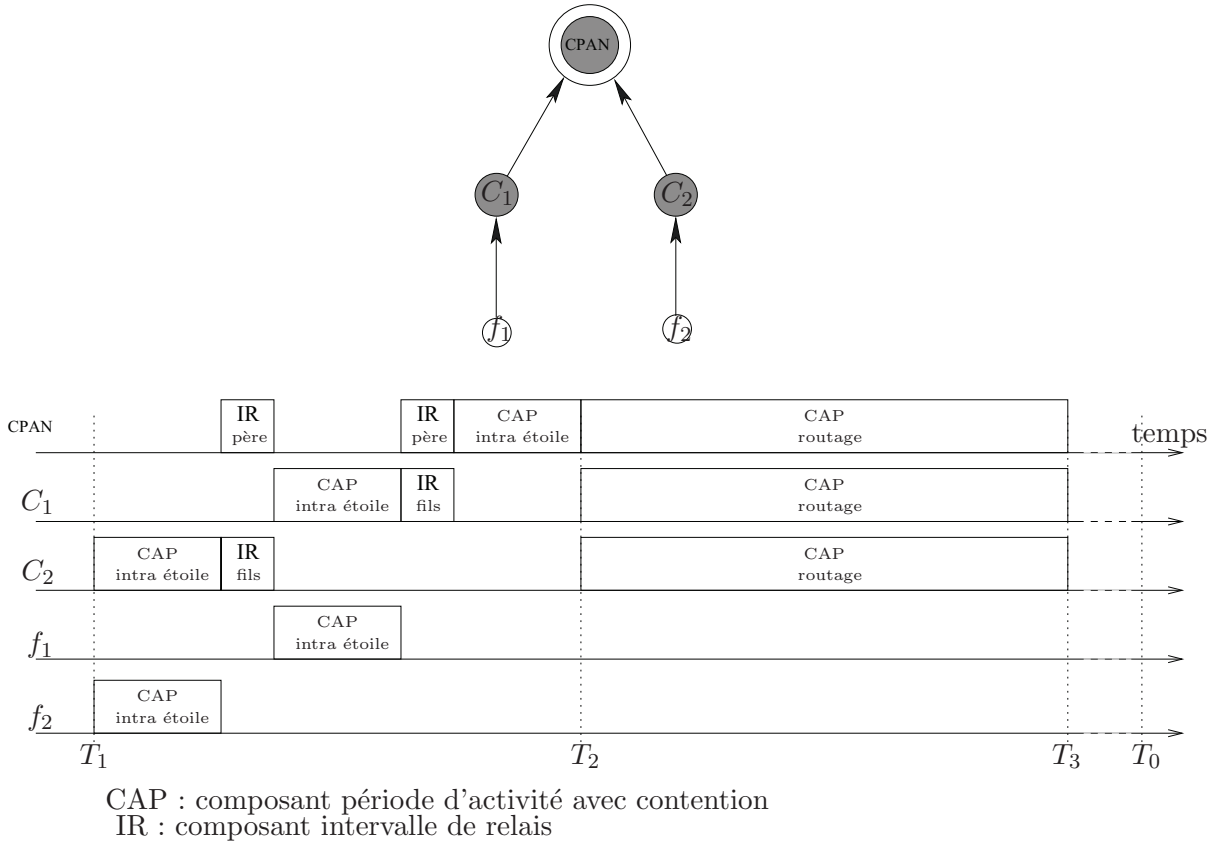


Figure 3.9 – Représentation des composants responsables des échanges de données durant l'intervalle $[T_1; T_3]$.

Les composants de routage, de construction, et de stockage des trames de données

L'ensemble de ces composants sont responsables du routage et de la gestion des trames de données.

Le composant *Routage hiérarchique* : Ce composant détermine le prochain saut selon l'algorithme de routage hiérarchique. Lorsque le nœud souhaite transmettre une

trame, le composant détermine le prochain saut à partir de l'adresse du destinataire et les paramètres L_m, R_m et C_m. Ensuite, le composant transmet son adresse au composant de *Construction de trame de données* en indiquant la période dans laquelle la trame doit être envoyée.

Le composant *Construction de trame de données* : Ce composant organise la construction des trames de données et l'assemblage en mémoire. Il réserve l'espace nécessaire (128 octets) puis renseigne les éléments de l'entête et la charge utile selon les informations fournies par le composant *Routage hiérarchique*. Ensuite, la trame assemblée est placée dans la file d'attente correspondante à la période d'activité pendant laquelle la trame devra être transmise.

Le composant *Gestion des files d'attente* : Ce composant gère cinq files d'attente dans lesquelles sont stockées les trames destinées à être transmises pendant une période donnée :

- une file d'attente pour la période intra-étoile (trafic non prioritaire),
- une file d'attente pour les trames de la période de routage $[T_2; T_3]$ (trafic non prioritaire),
- deux files d'attente pour les intervalles de relais père-fils (trafic prioritaire descendant),
- une file d'attente pour l'intervalle de relais fils-père (trafic prioritaire montant).

Pour rappel, cette implémentation supporte une topologie arborescente avec $R_m \leq 2$. Il y a donc (au plus) deux files d'attente correspondant au relais père-fils : une pour chaque fils coordinateur.

Chaque file ordonne les trames selon la méthode premier entré, premier sorti. Chaque file peut contenir jusqu'à sept trames de 128 octets. L'espace réservé pour les cinq files d'attente occupe donc 4480 octets, soit près de la moitié de la mémoire vive totale disponible (10240 octets) sur une carte TelosB.

La figure 3.10 représente les étapes lors d'un envoi ou un relais de trame. En fonction de la priorité du trafic (selon si la trame est envoyée en trafic prioritaire ou non prioritaire), le composant *Routage Hiérarchique* détermine le prochain saut. Ensuite, la trame

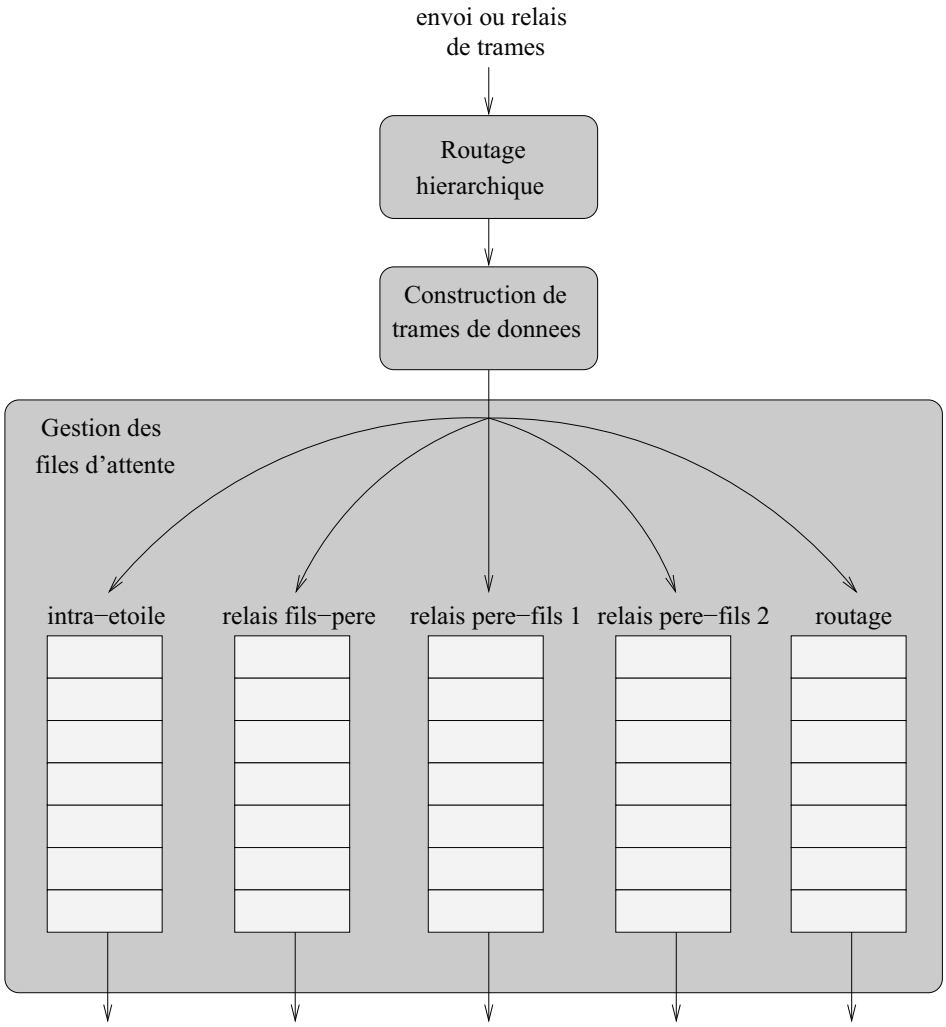


Figure 3.10 – Représentation de la répartition des trames dans les files d'attente.

est construite puis copiée en mémoire dans la file d'attente correspondant à la période d'activité durant laquelle elle sera transmise.

Les composants de configuration

Les composants de configuration sont regroupés dans le composant *MaCARI PIB* qui contient aussi tous les paramètres de configuration utiles à la couche MaCARI. Le composant *MaCARI PIB* contient également les tables de voisinage nécessaires au routage et à la détermination de la position du nœud dans l'arbre.

3.3 Adaptation et optimisation de l'implémentation de MaCARI

Le portage de MaCARI sur plateforme TelosB implique une phase d'ajustement afin d'optimiser le comportement de l'implémentation. Nous avons décidé de nous concentrer sur l'envoi et la synchronisation : les optimisations majeures ont été réalisées au niveau de la phase de chargement de trame dans le *buffer* d'envoi et au niveau de la période de synchronisation (cascade de *beacons*).

3.3.1 Phase de chargement de trame dans le *buffer* d'envoi du module radio

L'envoi d'une trame au niveau PHY se fait en deux phases : la phase de chargement dans le *buffer* d'envoi et la phase de transmission. Initialement, une trame à envoyer est stockée dans la mémoire RAM de la carte TelosB. La taille maximale d'une trame ne peut dépasser 128 octets. Pour transmettre une trame ainsi stockée, il est nécessaire de la copier à partir de la RAM dans le *buffer* d'envoi du module radio. La figure 3.11 représente l'architecture de la plateforme TelosB à ce niveau. Le microprocesseur et le module radio sont connectés par une liaison SPI (pour *Serial Peripheral Interface*), c'est-à-dire par un bus de donnée série synchrone opérant en *Full-Duplex* (le transport des données peut avoir lieu simultanément dans les deux sens).

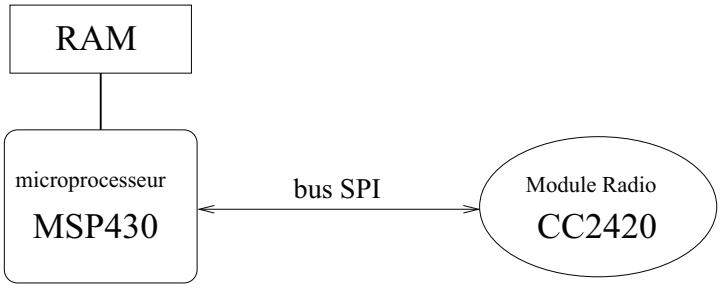


Figure 3.11 – Représentation du bus SPI reliant le module radio et le microprocesseur.

Sur TelosB, la fréquence des échanges sur le bus SPI est par défaut égale à 1 MHz, c'est-à-dire une microseconde par bit transmit. Une trame de taille maximale (128 octets, soit 1024 bits) doit donc être chargée en $1,024$ ms. Cependant, la transmission des données est perturbée au niveau du microprocesseur MSP430 [35] par une interruption à chaque fois qu'un octet a été transmis. Par conséquent, la durée de transmission de la trame est largement augmentée par le délai induit par le traitement des interruptions.

Dans l'implémentation de MaCARI, la durée de chargement dans le *buffer* d'envoi est mesurée à partir de l'horloge du microprocesseur, en comparant sa valeur lors du début de la tâche de chargement à sa valeur lorsque l'évènement de fin de la tâche est signalé. L'horloge du microprocesseur MSP430 est cadencée à 32768 Hz : elle permet une précision de l'ordre de $30,5$ microsecondes.

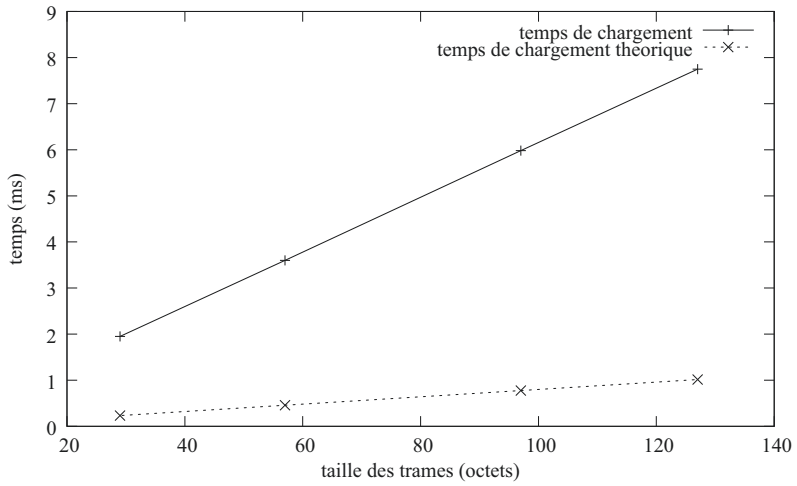


Figure 3.12 – Temps de chargement d'une trame dans le *buffer* d'envoi.

La figure 3.12 montre la durée des temps de chargement effectifs et théoriques dans le *buffer* selon la taille de la trame chargée. Pour le chargement effectif, chaque point

représente une moyenne de 10 mesures qui se sont révélées constantes (écart type nul) tandis que le temps de chargement théorique est déterminé analytiquement par la vitesse du bus SPI (1 MHz). Nous constatons que l'augmentation du temps de chargement est linéaire : le coefficient directeur de la courbe du chargement théorique vaut environ $0,008$ tandis que celui de la courbe du chargement réel vaut $0,067$. On peut en déduire que le temps de chargement effectif est environ $8,4$ fois plus important que le temps théorique.

La durée admise de chargement d'une trame de taille inférieure à 128 octets doit être proche de la milliseconde. Il existe trois possibilités pour accélérer cette phase.

La première possibilité est d'augmenter le nombre de bits transmis sans interruption. Il existe un mécanisme dans TinyOS qui permet de garantir l'atomicité d'une suite d'opérations en interdisant toute interruption. Grâce à cette propriété, on peut augmenter le nombre de bits transmis de manière atomique sur le bus SPI et ainsi réduire le délai dû aux interruptions du microprocesseur. Néanmoins cette opération a un impact important : en limitant la possibilité d'interruptions, une latence est répercutée sur toutes les tâches exécutées par le MSP430 pendant l'utilisation du bus SPI. Cette latence étant source d'erreurs et d'imprécisions, cette possibilité est donc écartée.

La seconde possibilité consiste à activer le mécanisme d'accès direct à la mémoire (DMA) sur le bus SPI. Le DMA [36] permet à un périphérique d'avoir accès à la RAM sans nécessiter l'intervention du microprocesseur. Cette méthode permet de libérer l'utilisation du CPU durant les transferts. L'envoi des données sur le bus SPI n'est donc pas perturbé de la même façon par les interruptions.

La figure 3.13 représente l'évolution du temps de chargement sans et avec DMA. Chaque point représente une moyenne de 10 mesures. L'écart type n'est pas représenté car il est négligeable. Nous observons que la durée de chargement avec le DMA activé est réduite. L'évolution moyenne de la durée selon la taille des trames est obtenue par régression linéaire selon la méthode des moindres carrés : l'équation de la droite est de la forme $y = 0,02 \times x + 0,96$ (avec y la durée en ms et x la taille de la trame en octets). Nous pouvons déduire que le transfert effectif des données avec DMA est environ 3 fois plus rapide que la durée de chargement sans DMA mais qu'il implique un délai constant supplémentaire de $0,96$ ms. Ce délai est attribué à la phase de configuration nécessaire au DMA de la plateforme pour envoyer les bits par le bus SPI [35].

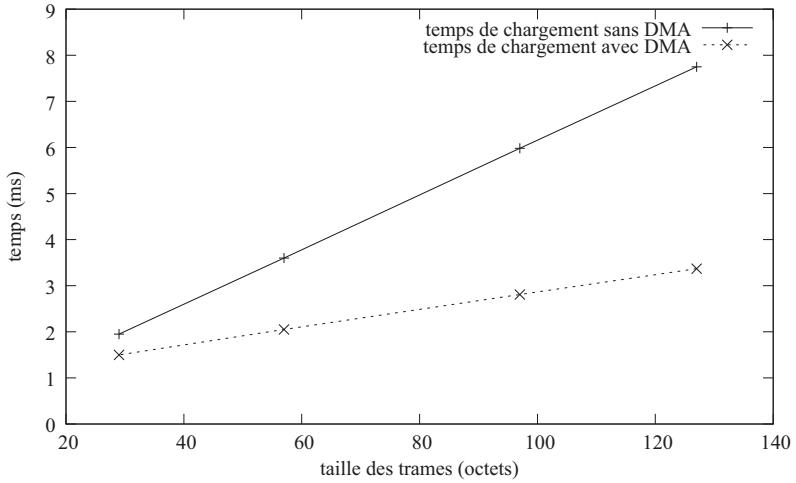


Figure 3.13 – Impact du DMA sur le temps de chargement dans le *buffer* d'envoi.

La troisième possibilité vise à augmenter la vitesse du bus SPI. Une augmentation de la fréquence de 1 MHz à 2 MHz s'opère en doublant la fréquence de l'horloge interne du microprocesseur ; cette opération entraine une hausse de la température du composant. L'horloge interne du MSP430 est un oscillateur contrôlé numériquement (DCO) cadencé à 4 MHz par défaut. Afin de calibrer l'oscillateur à la fréquence de 8 MHz sans surchauffe, le DCO de la version du MSP430 des cartes TelosB dispose d'une résistance externe (appelée Rosc) pour dissiper la chaleur (on peut commander son activation en paramétrant le jeu de composants propres au MSP430 de TinyOS).

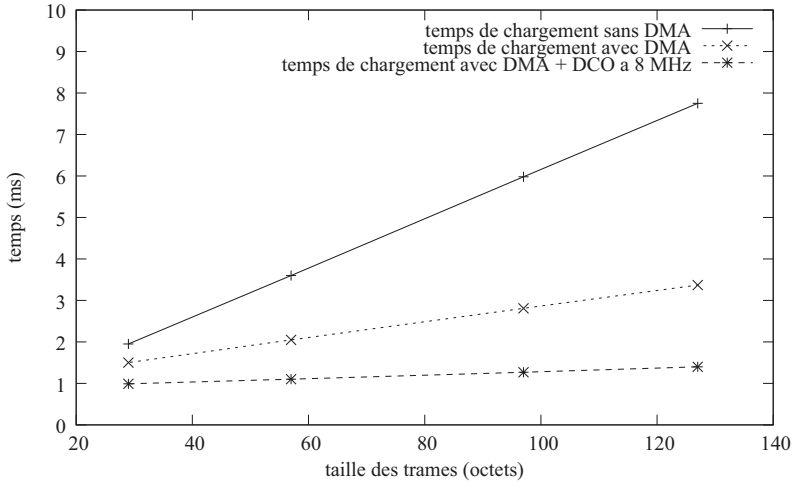


Figure 3.14 – Impact du cadencement du DCO à 8 MHz sur le temps de chargement.

La figure 3.14 représente l'impact du cadencement du DCO à 8 MHz sur le temps de chargement avec le DMA actif. Chaque point représente une moyenne de 10 mesures.

L'écart type n'est pas représenté car il est négligeable. Nous observons qu'avec le cadencement du DCO à 8 MHz, la durée de chargement est réduite. L'évolution moyenne de la durée selon la taille des trames avec DMA et DCO à 8 MHz est obtenue par régression linéaire selon la méthode des moindres carrés : l'équation de la droite est de la forme $y = 0,0043 \times x + 0,86$ (avec y la durée en ms et x la taille de la trame en octets). Le temps de transmission effectif des bits sur le bus SPI est multiplié par environ $4,7$ par rapport au temps de transmission effectif lorsque seul le DMA est activé (Le délai constant lié à la phase de configuration du DMA est lui aussi réduit à $0,86$ ms).

Bilan : Afin d'optimiser le temps de chargement dans le *buffer* d'envoi du module radio sur la plateforme TelosB, l'implémentation de MaCARI est configurée avec le DMA activé et le DCO cadencé à 8 MHz. Ce paramétrage permet d'obtenir un temps de chargement maximal de $1,42$ ms pour une trame de 128 octets contre 8 ms sans optimisation.

3.3.2 Période de synchronisation

La période de synchronisation est organisée à partir d'une cascade de *beacons* ; chaque nœud, hormis le CPAN, détermine l'instant T_0 du cycle à la réception d'un *beacon*. Par conséquent, chaque transmission de *beacon* doit être réalisée et se terminer à une date définie et contrôlée afin de garantir une synchronisation précise. La période de synchronisation est découpée dans l'implémentation en n intervalles de durée égale (où n est égal au nombre total de coordinateurs) appelés intervalles de cascade de *beacons*. La durée d'un intervalle doit permettre à un coordinateur de traiter un *beacon* et de le transmettre à son tour. La charge utile du *beacon* contient la liste ordonnée des adresses de tous les coordinateurs du PAN ; chaque adresse est codée sur deux octets. Selon la topologie du réseau, la taille de la charge utile du *beacon* dédié à cette liste d'adresses varie donc entre 2 et 90 octets (soit de 1 à 45 adresses de coordinateurs).

La durée de traitement, de chargement et de transmission de la trame de *beacon* varie selon sa taille. Ces trois phases (qui forment l'intervalle de cascade de *beacons*) sont illustrées dans la figure 3.15.

Pour minimiser la durée de la période $[T_0; T_1]$, il faut qu'un intervalle de cascade de *beacons* soit dynamiquement ajusté selon sa taille (elle même basée sur le nombre de

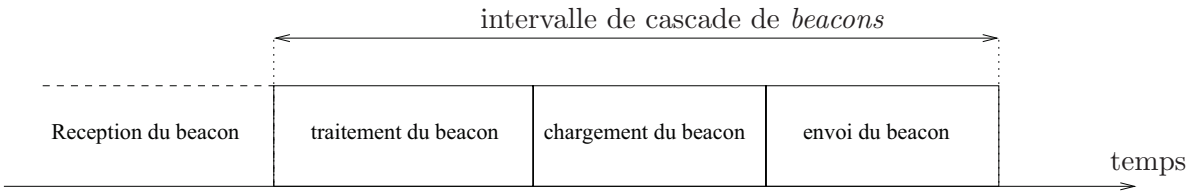


Figure 3.15 – Les trois phases qui composent un intervalle de cascade de beacons.

coordinateurs présents dans le réseau). Dans l'implémentation, les phases de l'intervalle de cascade de *beacons* sont des tâches de type non-prioritaire. Par conséquent, la durée des phases peut varier sensiblement notamment si ces tâches ont été ou non interrompues. L'objectif de notre optimisation est donc de déterminer une formule permettant d'évaluer la durée d'un intervalle de cascade de *beacons* en fonction du nombre de coordinateurs.

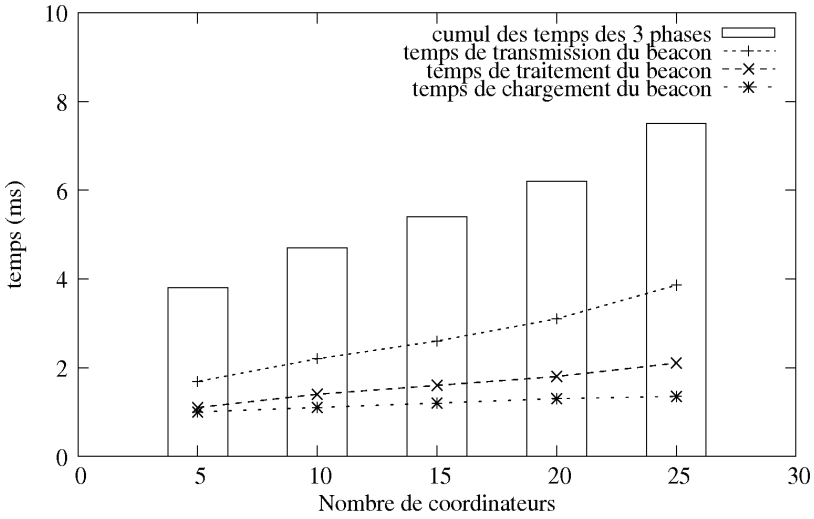


Figure 3.16 – Évaluation de la durée cumulée des trois phases délimitant un intervalle de cascade de *beacons*.

La figure 3.16 indique l'évolution de la durée maximale de chacune des phases selon le nombre de coordinateurs. Chaque point représente la borne supérieure de 10 mesures et non la moyenne car l'objectif est de déterminer une durée maximum. Les histogrammes représentent la durée cumulée maximum des trois phases pour un nombre de coordinateurs donné. On observe que chaque courbe a une forte tendance linéaire. Pour éviter des calculs polynomiaux lourds pour le microprocesseur, l'estimation de la durée est basée sur la fonction affine de la droite "tangente" aux valeurs de la durée cumulée des trois phases. Cette droite, qui n'est pas représentée pour alléger la figure, a pour équation $y = 0,18 \times x + 2,8$. On peut déduire de cette équation l'approximation de la durée d'un

intervalle de cascade de *beacons*, soit :

$$T_i = 0,18 \times nb_C + 2,8$$

avec T_i la durée d'un intervalle de cascade de *beacons* en ms et nb_C le nombre de coordinateurs.

L'unité de mesure du temps dans l'implémentation est le tic de d'horloge du microprocesseur, qui vaut $30,5$ microsecondes. L'équation utilisée dans le composant *Synchronisation* de l'implémentation de MaCARI pour l'évaluation de la durée d'un intervalle est la suivante :

$$T_i = 6 \times nb_C + 92$$

avec T_i la durée d'un intervalle de cascade de *beacons* en tics et nb_C le nombre de coordinateurs. Cette formule permet d'optimiser la durée d'un intervalle selon le nombre de coordinateurs et par conséquent la durée de la période de synchronisation $[T_0; T_1]$.

Chapitre 4

Évaluation

L'objectif de ce chapitre est d'évaluer les propriétés de notre implémentation au niveau de quatre points clés : la synchronisation, les méthodes d'accès, le délai de bout en bout et la consommation énergétique.

4.1 Synchronisation

L'évaluation de la synchronisation a pour but de déterminer la précision de la synchronisation dans un réseau MaCARI selon la position d'un nœud. MaCARI est un protocole reposant sur une référence de temps commune à toutes les entités et basée sur le temps du CPAN : une synchronisation précise est nécessaire pour éviter les collisions de trames lors de l'accès au médium et le chevauchement des périodes d'activité.

La qualité de la synchronisation est liée à la détermination de la date de début de cycle T_0. Pour déterminer précisément T_0, chaque entité se base sur l'instant de réception d'un *beacon*. Or l'estimation de cette date est approximative car l'exécution de la tâche d'envoi d'un *beacon* se fait à un instant qui peut varier sensiblement (en fonction des évènements matériels par exemple). L'impact de cette variation est d'autant plus grand que le nœud est loin dans l'ordre de la synchronisation (et donc loin de la racine de l'arbre). Par conséquent, dans le cadre de la cascade de *beacons* pendant laquelle les transmissions

de *beacons* sont successives, nous devons étudier l'impact de la position du nœud sur la qualité de la synchronisation. D'autre part, une latence dans l'enchaînement des tâches et la dérive des horloges peuvent entraîner un décalage de synchronisation entre les nœuds au cours du cycle.

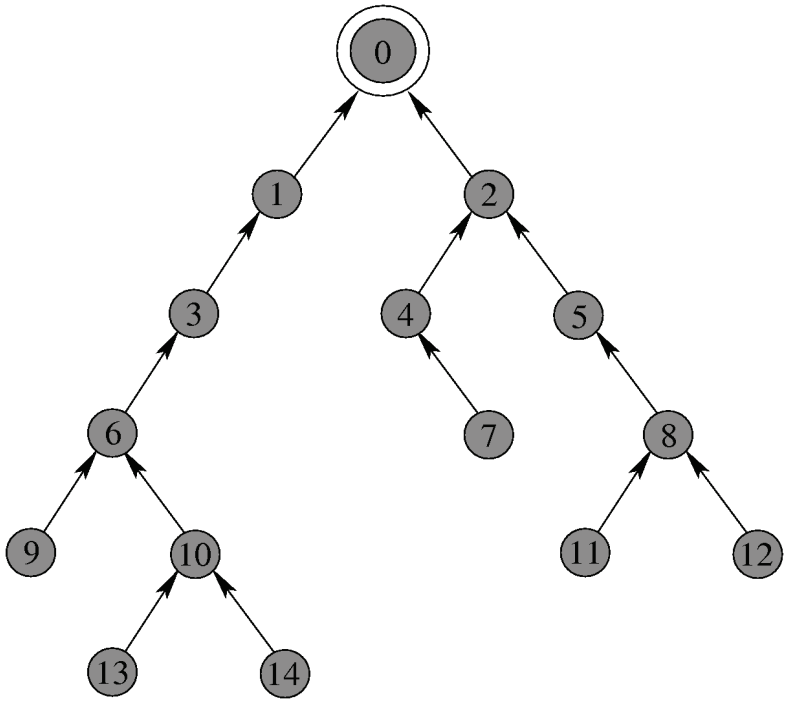


Figure 4.1 – Représentation de la topologie arborescente utilisée pour évaluer la synchronisation.

Pour mesurer la qualité de la synchronisation de l'implémentation, nous avons étudié une topologie de réseau MaCARI en arbre binaire composée d'un CPAN et de quatorze coordinateurs, illustrée par la figure 4.1. Chaque nœud est numéroté de manière croissante selon sa position dans la cascade de *beacons*.

Les expérimentations suivantes ont été réalisées avec la contribution de S. Roicomte, stagiaire de l'IUT de Clermont-Ferrand, qui a construit un dispositif capable d'accueillir les cartes TelosB et de les connecter à un analyseur d'états logiques [37]. Pour chaque carte connectée à cette installation, nous étudions les variations de l'état logique d'une LED dont l'état haut correspond à une période d'activité. Ainsi, nous pouvons évaluer la précision de la synchronisation en mesurant les différences de temps entre les entités lors d'un changement d'état pour la même date du cycle. Néanmoins, pour relier une carte au dispositif, il faut réaliser des opérations de soudure précises. Ces opérations ont

été effectuées sur 4 cartes TelosB. Par conséquent, nous pouvons réaliser des mesures simultanément sur quatre nœuds de l'arbre.

Pour évaluer la qualité de la synchronisation dans un réseau MaCARI, nous proposons d'étudier l'instant T_2 d'un cycle, qui est une date commune à tous les coordina-teurs correspondant au début de la période de routage. Une interface graphique fournie avec l'analyseur d'états logiques permet de visualiser les changements d'état des entrées observées (le pilotage des LED).

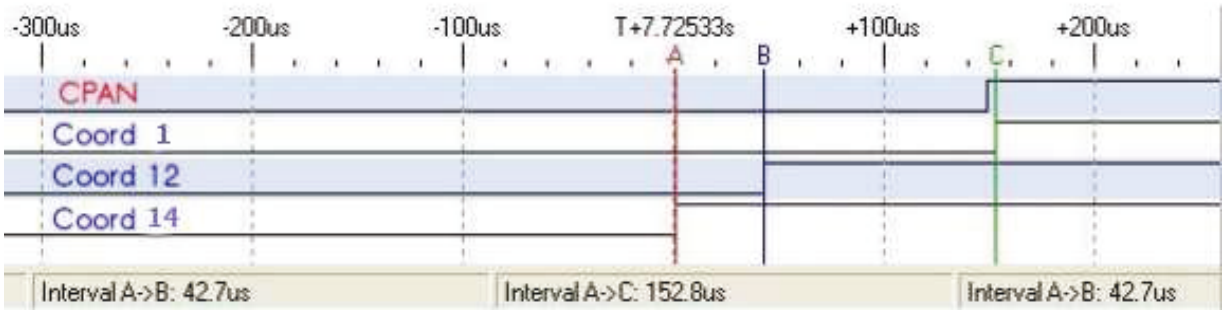


Figure 4.2 – Exemple de mesures réalisées avec l'analyseur d'états logiques sur la précision de la synchronisation dans un réseau MaCARI.

La figure 4.2 propose un exemple de mesures avec l'analyseur d'états logiques, *via* l'interface graphique. Cette figure représente les décalages relevés entre le CPAN et les coordinateurs 1, 12 et 14 pour la topologie illustrée dans la figure 4.1 sur l'instant T_2 pour un cycle d'une durée totale d'environ $3,6$ secondes. Pour cette mesure, on observe un écart maximal de précision (appelé déviation) sur la date T_2 entre les nœuds (borné sur la figure par les lignes curseurs A et C) de $152,8$ μs.

La figure 4.3 montre la différence entre l'instant T_2 du CPAN et l'instant T_2 pour les coordinateurs 1, 12 et 14 de la topologie en arbre illustrée dans la figure 4.1. Rappelons que les nœuds sont ordonnés selon un parcours en largeur de l'arbre pour la cascade de *beacons*. Cette figure présente la déviation par rapport au temps de référence du CPAN pour des cycles pour lesquels l'intervalle $[T_0; T_2]$ vaut environ $0,5$ seconde, $0,9$ seconde, $1,8$ secondes, $3,6$ secondes et $7,1$ secondes. Chaque point représente la moyenne de 28 mesures (l'écart type n'est pas représenté pour des raisons de lisibilité). Plus les mesures sont proches de 0 (valeur correspondant à une déviation nulle), plus la qualité de la synchronisation est bonne. Nous pouvons faire trois observations à partir de cette fi-

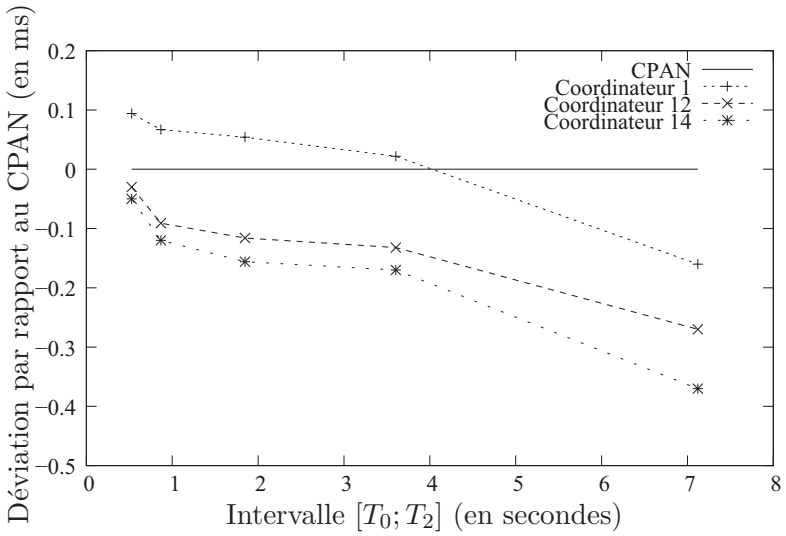


Figure 4.3 – Évolution de la synchronisation par rapport au CPAN en fonction de la durée du cycle, pour la topologie en arbre illustrée sur la figure 4.1.

gure. D'abord, on observe que la qualité de la synchronisation se dégrade sensiblement plus l'intervalle $[T_0; T_2]$ est grand. Nous pouvons expliquer ce phénomène par un nombre d'interruptions des tâches plus important lorsque la durée de l'intervalle grandit, mais également par le décalage naturel de l'oscillateur à quartz du CC2420 pouvant varier de ± 40 μs par seconde (pour $7,1$ secondes, l'écart d'horloge peut valoir jusqu'à ± 384 μs). D'autre part, nous observons que les nœuds positionnés en fin de cascade ont une synchronisation moins précise. Cette tendance est explicable par les latences induites par le cumul d'imprécisions lors d'envois successifs de *beacons* pendant la période de synchronisation. Enfin, on peut constater que pour une durée $[T_0; T_2]$ inférieure à 4 secondes, les déviations moyennes mesurées sont inférieures à 200 μs pour tous les nœuds de l'arbre, soit moins d'une période de *backoff* (valant 320 μs).

Bilan : La synchronisation dans l'implémentation de MaCARI sur TelosB est précise : les déviations de synchronisation de cette l'implémentation de MaCARI sont en moyenne inférieures à 200 μs partout dans l'arbre (topologie de 14 nœuds) pour un cycle dont la durée est inférieure à $3,6$ secondes. Néanmoins, pour éviter que les périodes d'activités ne se chevauchent, chaque période d'activité est séparée par une période de garde dont la durée est égale à un *backoff* de 320 μs.

4.2 Les méthodes d'accès

Chaque cycle MaCARI contient une période d'activité dans laquelle sont utilisées deux méthodes d'accès au médium : un accès avec CSMA/CA slotté et un accès sans. L'objectif de cette partie est d'étudier les performances en termes de débit selon la méthode d'accès. Le débit maximum théorique est dépendant de la modulation et de la bande de fréquences utilisées. Il est impacté par les mécanismes de la méthode d'accès utilisée et par la taille des trames qui composent le trafic (que nous supposons uniforme).

Nous étudions tout d'abord la durée de la phase de transmission physique des trames et le débit potentiel de l'implémentation associé. Puis, nous présentons les performances des deux méthodes d'accès.

4.2.1 Phase de transmission physique des trames

La phase de transmission physique d'une trame est commune à toutes les méthodes d'accès. Elle est réalisée au niveau de la couche physique : elle débute lors du déclenchement de la commande de transmission d'une trame déjà chargée dans le *buffer* d'envoi du composant radio et se termine lorsque l'évènement de fin de transmission est signalé par la couche physique de l'émetteur.

Pour comparer la durée de cette phase dans l'implémentation de MaCARI avec la durée théorique selon la taille des trames, nous avons mesuré cette période avec l'horloge du microprocesseur MSP430 (d'une précision de $30,5$ μs). Les mesures ont été validées en relevant les temps de réception des trames avec un *sniffer*. Un *sniffer* est une entité réseau qui récupère à la volée les trames échangées dans le réseau.

La figure 4.4 compare la durée théorique de l'envoi et le temps réellement mesuré fourni par l'implémentation de MaCARI, sur un nœud TelosB, en fonction de la taille de la trame envoyée (avec un pas de trente octets environ). Chaque point représente une moyenne de dix mesures et l'écart type (quasi nul) n'est pas représenté.

La figure 4.4 montre que la durée de la phase de transmission physique en fonction de la taille de la trame est linéaire. L'équation linéaire du temps réel mesuré obtenue par régression linéaire ($y = 0,03 \times x + 0,767$) a le même coefficient directeur que l'équation du temps théorique ($y^* = 0,3 \times x$). Cependant, on constate qu'il existe un délai constant

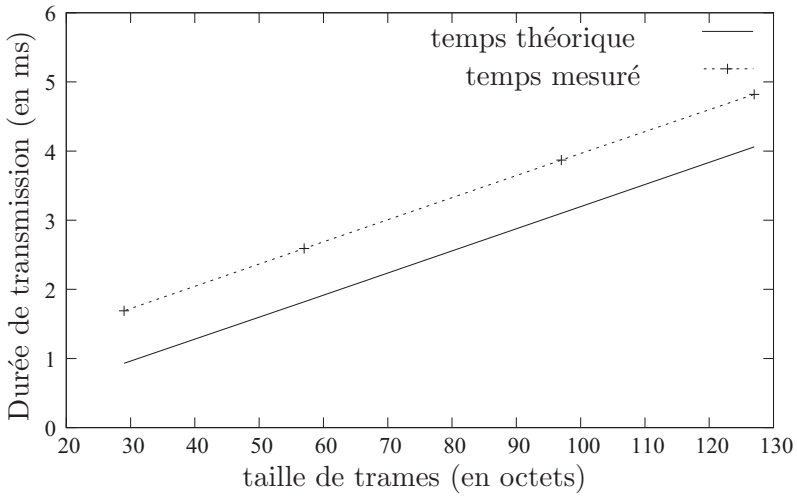


Figure 4.4 – Durée de transmission effective en fonction de la taille des trames envoyées.

de $0,767$ ms dû à la gestion du module radio par le système d'exploitation TinyOS. Ce délai constant induit une baisse du débit potentiel par rapport au débit théorique et ce, indépendamment de la méthode d'accès.

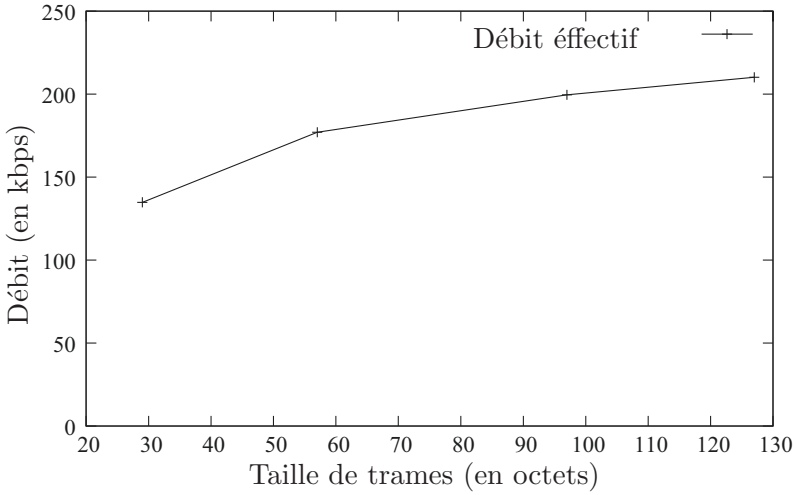


Figure 4.5 – Débit en fonction de la taille des trames envoyées.

La figure 4.5 montre le débit effectif en fonction de la taille des trames envoyées. Cette courbe est obtenue en calculant le débit en kbps à partir des mesures issues de la figure 4.4 précédente. On constate une différence des performances observées avec les performances théoriques. Rappelons que le module radio CC2420 offre un débit théorique de 250 kbps. Le délai constaté dans la figure 4.4 provoque une réduction du débit, ceci est particulièrement préjudiciable quand les trames sont petites. Pour des trames courtes de 29 octets, le débit est de $137,5$ kbps soit inférieur de 45% au débit théorique. Pour

des trames longues de 128 octets, nous constatons un débit de 210, 5 kbps, soit inférieur de 15, 8% au débit théorique.

L'étude des méthodes d'accès doit prendre en compte les limites définies par la phase de transmission physique.

4.2.2 Envoi direct

L'envoi direct correspond à la phase de transmission lorsque le médium est supposé libre. L'envoi direct est un envoi de trame sans test CCA.

MaCARI utilise l'envoi direct pour les acquittements dans chaque période d'activité. Pour les trames de données, l'envoi direct est utilisé dans trois périodes différentes : durant la phase de synchronisation pour l'envoi du *beacon*, durant la période d'activité intra-étoile pour les GTS (ce service n'est pas implémenté dans cette version) et durant les intervalles de relais. Les performances de cette méthode d'accès correspondent à celles évaluées en termes de débit dans la partie 4.2.1.

4.2.3 Envoi avec CSMA/CA slotté

Dans MaCARI, le CSMA/CA slotté est utilisé pour les périodes pour lesquelles les stations sont en concurrence pour émettre : durant la période d'activité intra-étoile et durant la période de routage $[T_2; T_3]$. L'évaluation qui suit est basée sur le CSMA/CA slotté avec les paramètres suivants : BE égal à 2 (avec BE minimum égal à 2 et BE maximum égal à 5), $NB = 0$ et nombre d'essais de renvoi de trame égal à 3.

4.2.3.1 Débit maximum

Dans un premier temps, nous avons évalué le débit maximal atteint dans les conditions optimales où l'émetteur TelosB envoie des trames en CSMA/CA slotté sans que le canal ne soit occupé. Pour ce faire, nous avons mesuré la durée de la phase de transmission au niveau MAC, c'est à dire entre l'appel de l'algorithme du CSMA/CA slotté et le signalement de fin de transmission envoyé par la couche physique de l'émetteur. Nous en avons déduit le débit maximum illustré sur la figure 4.6, dans laquelle chaque point

indique une moyenne basée sur une répétition de dix mesures (l'écart type n'étant pas représenté).

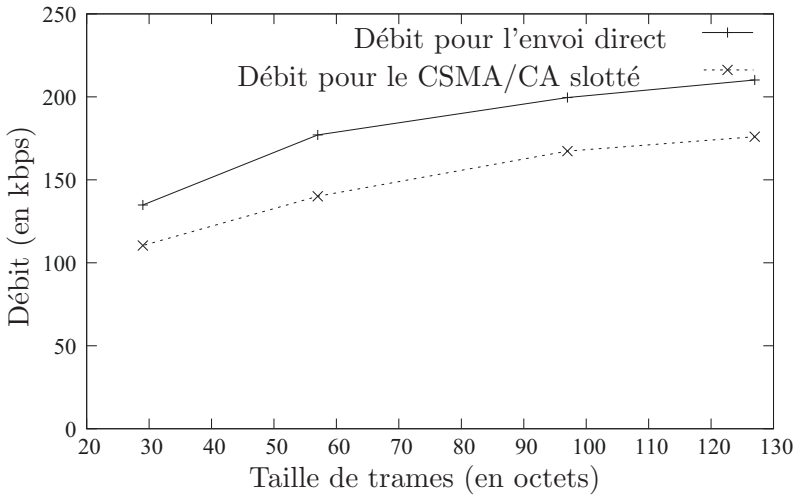


Figure 4.6 – Évaluation du débit maximal selon la méthode d'accès.

Comme le montre la figure 4.6, le débit maximum de l'implémentation de MaCARI pour l'envoi en CSMA/CA slotté est de 168 kbps pour un trafic composé de trames de 128 octets, débit inférieur de $21,5\%$ par rapport au débit pour une taille de trame équivalente en envoi direct. Cette différence d'efficacité est due à la présence de deux CCA et au mécanisme de *backoff* (qui, pour $BE = 2$, induit un délai aléatoire compris entre 0 et 7 périodes de *backoff*).

4.2.3.2 Période de collecte

Lors de la période de collecte dans MaCARI, les feuilles sont en concurrence pour émettre leurs trames aux coordinateurs. L'efficacité du CSMA/CA slotté est dépendante du volume du trafic et du nombre de feuilles qui veulent émettre. Nous avons disposé une topologie en étoile composée de six feuilles et d'un coordinateur, comme illustré sur la figure 4.7.

Pour valider les performances de la méthode d'accès, nous avons reproduit l'expérience décrite dans [8] pour évaluer le comportement de l'implémentation pour la période de collecte. Pour cela, nous avons fait varier le nombre de feuilles actives et la quantité de trafic générée par chaque feuille (1, 3 et 5 trames avec acquittements). Nous évaluons le nombre de trame reçues par le coordinateur. Les trames font 50 octets et la puissance

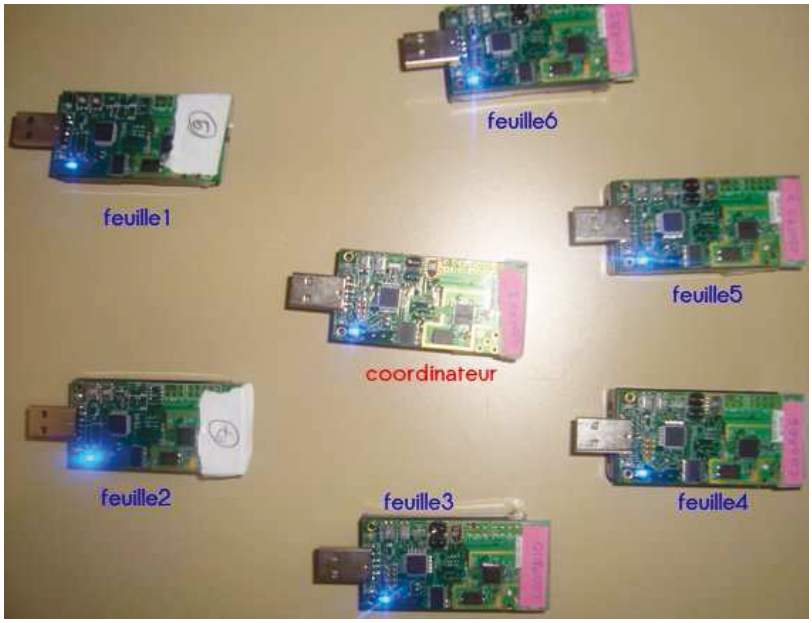


Figure 4.7 – Photographie de cartes TelosB formant une topologie en étoile.

d'émission de l'ensemble des entités est fixée à 0 dBm. Les nœuds sont disposés comme illustré sur la figure 4.7. Ensuite, après une période de synchronisation, chaque feuille tente d'envoyer ses trames au même instant. Chaque point correspond à une moyenne de 20 répétitions (l'écart type n'est pas représenté).

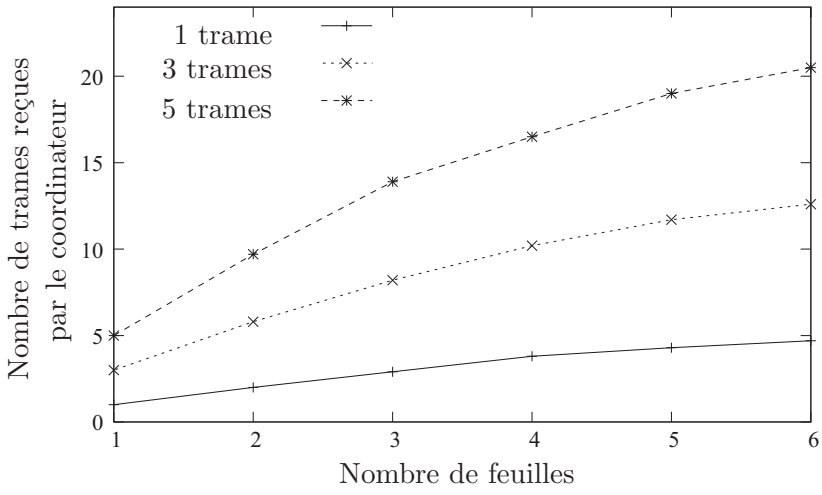


Figure 4.8 – Nombre de trames reçues par le coordinateur en fonction du nombre de feuilles et du nombre de trames émises par chaque feuille (trames de 50 octets avec acquittement).

La figure 4.8 montre que lorsque les feuilles émettent une trame seulement, des pertes de trames sont constatées à partir de 5 feuilles actives. Pour un trafic de 3 et 5 trames

par feuille, nous observons des pertes de trames dès 3 feuilles. Ce phénomène s'explique par l'apparition de collisions de trames et d'échecs de transmission (trame non envoyée au bout de trois essais). Le comportement de l'implémentation est en adéquation avec les résultats de simulations (réalisés avec le même protocole expérimental) et donnés sur la figure 4.11 de [8].

4.2.3.3 Période de routage

Lors de la période de routage, les coordinateurs sont en concurrence entre eux pour transmettre les trames du trafic non prioritaire à destination d'un autre nœud du réseau. Contrairement à la période de collecte où les feuilles ne font que transmettre et le coordinateur ne fait que recevoir les trames (pour un réseau dont les feuilles sont uniquement des capteurs), la période de routage nécessite que les coordinateurs effectuent des relais de trames impliquant l'alternance de phases d'émission et de réception.

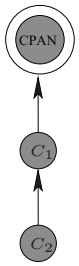


Figure 4.9 – Topologie linéaire simple pour évaluer la période $[T_2; T_3]$.

Ici, notre objectif est de vérifier le comportement de l'implémentation pour une topologie linéaire simple, composée d'un CPAN et de deux coordinateurs, illustré sur la fi-gure 4.9. À l'instant T_2, chaque coordinateur envoie une trame sans acquittement en CSMA/CA slotté en direction du CPAN (nous n'utilisons pas volontairement les acquittements afin de mieux mettre en évidence les envois infructueux). La trame envoyée par le coordinateur C_2 doit donc être relayée par le coordinateur C_1 pour atteindre le CPAN. L'algorithme de routage utilisé est le routage hiérarchique. Nous avons effectué 10 répétitions de cette expérience dans le but d'étudier le nombre de trames reçues par le CPAN pendant une durée suffisamment grande pour permettre à toutes les trames générées par chacun des coordinateurs d'être acheminées vers le CPAN. Nous avons placé un *sniffer* afin de contrôler la chronologie de la transmission effective des trames.

Si la trame de C_1 est reçue dans 100% des cas, la trame de C_2 n'est reçue que dans 50% des cas environ. Pour comprendre ce comportement inattendu, nous avons examiné plus en détail les trames reçues et envoyées par C_1. Nous constatons que la trame envoyée par C_2 n'est pas reçue par le CPAN lorsque C_2 émet sa trame avant C_1. Dans ce cas, le *sniffer* capte bien la trame de C_2 mais celle-ci n'est ensuite pas relayée par C_1. Ce phénomène révèle un problème d'ordre conceptuel ayant un impact fort sur les performances durant cette période.

Analyse du problème : MaCARI utilise l'algorithme de CSMA/CA slotté de la norme IEEE 802.15.4 (vu en 2.1.1) lors de la période de routage. À l'instant T_2, chacun des coordinateurs tente de transmettre ses trames en instance en CSMA/CA slotté : la prochaine frontière de période de *backoff* est localisée et chaque coordinateur tire un *backoff* aléatoirement dans l'intervalle $[0; 2^{BE} - 1]$ avant d'effectuer un premier CCA. Si ce CCA détecte le canal comme étant occupé, la procédure complète est renouvelée avec un BE plus grand.

La norme ne précise pas la politique à adopter pour l'état de la radio pendant la durée du *backoff*. Il existe deux possibilités : soit le composant radio est placé en mode réception, soit il est placé en mode sommeil. Dans le premier cas, la réception de trames est possible pendant la phase d'émission (il n'y a pas d'économie d'énergie pendant le *backoff*). Dans le second cas, la réception de trames est impossible pendant la phase d'émission (le nœud économise de l'énergie durant le *backoff*).

Le CSMA/CA slotté du standard IEEE 802.15.4 est un algorithme qui est supposé minimiser la durée d'écoute du médium en réalisant des CCA ponctuellement plutôt que continuellement comme dans le standard IEEE 802.11 [38]. Le diagramme présentant l'algorithme dans figure 2.4 du chapitre 2 ne mentionne à aucun moment l'éventualité d'une réception. Cette constatation permet de renforcer l'hypothèse pour laquelle le composant radio est considéré en sommeil durant un *backoff* afin d'économiser de l'énergie. L'interprétation faisant de CSMA/CA slotté une méthode d'accès pensée pour une étoile de capteurs émetteurs dont le cœur est un coordinateur récepteur est privilégiée. L'implémentation TKN15.4 se base sur cette déduction et place la radio en sommeil pendant les *backoffs*. Par conséquent, lorsqu'un coordinateur tente de transmettre une trame, il ne

peut recevoir aucune trame tant que la trame n'est pas transmise ou détruite. Cette configuration priorise la transmission de trames et l'économie d'énergie au détriment de la réception.

Pour la période de routage de MaCARI, cette configuration entraîne, comme nous allons le voir, des transmissions infructueuses systématiques et donc une occupation inutile du médium (augmentant avec le volume du trafic). L'expérience réalisée pour la topologie illustrée sur la figure 4.9 révèle ce phénomène : si le coordinateur C_2 a accès au médium avant C_1 pour envoyer sa trame, C_1 se trouve en période de *backoff* et ne peut recevoir aucune trame. La trame de C_2 est donc transmise à perte. Par extension, plus les coordinateurs ont de trames à transmettre, moins le relayage de trames est efficace.

Proposition : Il n'existe pas à l'heure actuelle d'expérimentation de MaCARI visant à évaluer les performances du CSMA/CA slotté à forte charge lors de la période de routage [1].

Néanmoins, il semble préférable de prioriser la réception sur la transmission durant cette période. Cela implique une gestion de la politique de réception durant les *backoffs* : si une trame est reçue durant un *backoff*, la trame en cours de transmission est-elle annulée, suspendue, ou retransmise (avec quels paramètres BE, NB et CW) ?

La modification de l'algorithme pour la période de routage dans MaCARI n'est pas l'objet de ce mémoire. La période de routage $[T_2; T_3]$ de l'implémentation utilise donc l'algorithme de CSMA/CA slotté dans lequel le composant radio est placé en sommeil lors des *backoffs*, conformément à l'algorithme utilisé dans les simulations et décrit dans [8] (algorithme 7 page 110) priorisant ainsi la transmission et l'économie d'énergie au dépend d'un relayage optimal.

4.3 Délai de bout en bout

Cette évaluation vise à déterminer analytiquement le délai de bout en bout pour le trafic prioritaire, selon sa destination et la durée des différentes périodes de MaCARI.

1. car : (i) le protocole MAC de cette période n'est pas forcement le CSMA/CA slotté dans OCARI (utilisation du TDMA), (ii) protocole de routage EOLSR n'est pas implémenté actuellement dans le simulateur.

Pour cela, nous supposons que tout le trafic prioritaire est adapté au dimensionnement des périodes d'activité de l'arbre, c'est-à-dire que ces périodes ne sont jamais saturées. Nous devons considérer le délai de bout en bout dans le pire cas possible : une feuille génère une trame en fin de période d'échange intra-étoile mais n'a pas assez de temps pour l'envoyer, ce qui provoque un phénomène de report et nécessite l'attente d'un cycle entier avant que la trame ne soit remontée au CPAN. Pour rappel les GTS ne sont pas implémentés, donc nous supposons que les trames envoyées en trafic prioritaire par les feuilles sont reçues à coup sûr par le coordinateur de leur étoile.

D'abord nous déterminons la formule permettant de quantifier la durée d'un cycle MaCARI puis nous déduisons le délai de bout en bout selon la destination du trafic.

4.3.1 Durée d'un cycle MaCARI

Un cycle MaCARI est composé de quatre périodes dont la durée est paramétrable. La durée d'un cycle varie selon les paramètres de durée de chaque période d'activité et le nombre de coordinateurs. Pour l'intervalle $[T_1; T_3]$, l'unité de temps utilisée est le slot d'activité MaCARI. Un slot de base de MaCARI fait 60 symboles (3 périodes de *backoff*) soit 960 μs. L'exposant de durée d'un slot (EDS), l'équivalent du *Superframe Order* du standard IEEE 802.15.4, permet de dimensionner la durée des slots d'activité d_{slot} comme suit :

$$d_{slot} = 2^{EDS} \times 960 \times 10^{-3}.$$

Période de synchronisation $[T_0; T_1]$: La durée de la période de synchronisation dépend du nombre de coordinateurs nb_C et de la durée T_b d'un intervalle d'envoi de *beacon* exprimée en ms (optimisée dans 3.3.2). T_b est tel que :

$$T_b = 0,18 \times nb_C + 2,8.$$

$$d_{[T_0;T_1]} = nb_c \times T_b.$$

On peut donc calculer la durée de la période de synchronisation selon le nombre nb_C de coordinateurs avec la formule suivante :

$$d_{[T_0;T_1]} = nb_C \times (0,18 \times nb_C + 2,8).$$

Période $[T_1;T_2]$: La durée de $[T_1;T_2]$ dépend de la durée de la période d'activité de chaque étoile et du nombre d'étoiles du réseau. La durée de la période d'activité d'une étoile est la même pour toutes les étoiles du réseau : elle est composée d'une période intra-étoile et d'un intervalle de relais. La durée de la période intra-étoile d_{intra}, exprimée en ms, est paramétrable en fixant nb_S le nombre de slots d'activité pour cet intervalle (selon la charge générée par l'ensemble des feuilles). d_{intra} est tel que :

$$d_{intra} = nb_S \times d_{slot}.$$

La durée de l'intervalle de relais d_{relais}, exprimée en ms, est paramétrable en fixant nb_S le nombre de slots d'activité pour cet intervalle, selon la charge du trafic prioritaire :

$$d_{relais} = nb_S \times d_{slot}.$$

Une période de garde égale à deux périodes de *backoff* (soit d_{garde} égal à 640 μs) précède chaque période d'activité pour permettre les transitions entre chaque intervalle. La durée de la période d'activité d'une étoile d_{etoile}, exprimée en ms, compte deux périodes de garde : une avant la période intra-étoile et une avant la période de relais. d_{etoile} est calculée comme suit :

$$d_{etoile} = d_{intra} + d_{relais} + 2 \times d_{garde}.$$

La durée de la période $[T_1;T_2]$, exprimée en ms, dépend du nombre d'étoiles (équivalent au nombre de coordinateurs nb_C) et de la durée de la période d'activité d'une étoile d_{etoile} soit :

$$d_{[T_1;T_2]} = d_{etoile} \times nb_C.$$

Période de routage $[T_2;T_3]$: La durée de $[T_2;T_3]$, exprimée en ms, dépend de la charge du trafic non prioritaire. $[T_2;T_3]$ est paramétrable en fixant nb_{S_R} le nombre de

slots d'activité pour la période de routage et compte deux périodes de garde. Nous avons donc :

$$d_{[T_2;T_3]} = nb_{S_R} \times d_{slot} + 2 \times d_{garde}.$$

Période de sommeil $[T_3; T_0]$: La durée de $[T_3; T_0]$ détermine la période d'inactivité pendant laquelle toutes les entités du réseau sont en sommeil.

Durée d'un cycle : La durée d'un cycle MaCARI d_{cycle}, exprimée en ms, est déterminée en additionnant la durée des quatre périodes, soit :

$$d_{cycle} = d_{[T_0;T_1]} + d_{[T_1;T_2]} + d_{[T_2;T_3]} + d_{[T_3;T_0]}.$$

4.3.2 Trafic prioritaire à destination du CPAN

Dans le cadre d'un réseau de collecte, le trafic est remonté vers le CPAN. Dans ce contexte, l'ordonnancement des périodes d'activité des étoiles est orienté pour favoriser le trafic montant. Le pire des cas pour ce type d'arbre est défini par le cas suivant :

- la trame est générée par une feuille qui appartient à l'étoile dont la période d'activité est la première dans $[T_1; T_2]$,
- la trame est reportée (elle est envoyée lors du cycle global suivant),
- le dernier saut en relais garanti vers le CPAN est effectué durant le dernier intervalle de relais de $[T_1; T_2]$.

Dans ce cas, le délai maximum d'acheminement de la trame d est inférieur à la durée cumulée d'un cycle global et d'une période $[T_1; T_2]$ soit :

$$d \leq d_{cycle} + d_{[T_1;T_2]}.$$

4.3.3 Trafic prioritaire à destination quelconque

Nous supposons à présent que le trafic généré par les feuilles a pour destination une feuille quelconque et que le séquencement des périodes d'activité est alterné entre un

séquencement favorisant le trafic montant et un séquencement favorisant le trafic descendant.

Le pire des cas, abordé dans 2.2.8, indique que le délai d'acheminement d'une trame reportée de bout en bout est de 4 cycles maximum si la destination est quelconque. Nous avons donc :

$$d \leq 4 \times d_{cycle}.$$

4.4 Consommation énergétique

Cette évaluation vise à déterminer analytiquement la consommation énergétique de cette implémentation de MaCARI pour chaque type de nœud : les coordinateurs (le CPAN est considéré comme un coordinateur) et les feuilles. Une étude expérimentale visant à étudier l'autonomie est hors du cadre de ce mémoire car elle est davantage liée aux propriétés des batteries utilisées (recherches menées par le LRI, partenaire du projet OCARI) qu'au modèle de cartes utilisé.

Nous proposons de déterminer la durée pendant laquelle le module radio est actif, notée Δ, lors d'un cycle MaCARI. L'activité radio pour chaque type de nœud au cours d'un cycle MaCARI varie selon la période comme illustré dans le tableau 4.1.

Période $[T_0; T_1]$: La consommation énergétique d'un nœud pendant $[T_0; T_1]$ dépend de sa position dans l'ordonnancement défini dans la cascade de *beacons* : plus la position du nœud est éloignée de celle du CPAN, plus la durée d'écoute sera grande. Par conséquent, on étudiera la durée moyenne pendant laquelle un type de nœud active son module radio durant cette période.

Le module radio d'une feuille est inactif dès qu'elle termine la réception du *beacon* du coordinateur auquel elle est associée. Par conséquent, la durée moyenne pendant laquelle le module radio est actif pour une feuille, notée δ_f, est estimée à la moitié de la durée de la période $[T_0; T_1]$ soit :

$$\delta_f[T_0; T_1] = \frac{d_{[T_0;T_1]}}{2}.$$

Tableau 4.1 – État du module radio au cours d'un cycle MaCARI selon la période.

Périodes d'un cycle MaCARI	**Coordinateur**	**Feuille**
$[T_0; T_1]$ (période de synchronisation)	ACTIF pendant la moitié de la période + la durée d'envoi d'un beacon	ACTIF pendant la moitié de la période
$[T_1; T_2]$ (périodes d'activité)	ACTIF pendant la période d'activité de son étoile et les relais avec ses coordinateurs fils	ACTIF pendant la période d'échanges intra-étoile
$[T_2; T_3]$ (période de routage)	ACTIF toute la période	INACTIF
$[T_3; T_0]$ (période d'inactivité)	INACTIF	INACTIF

Un coordinateur a le même comportement d'écoute qu'une feuille mais doit en plus transmettre son *beacon*. Dans l'implémentation, le module radio d'un coordinateur est placé en sommeil entre la réception du *beacon* de son coordinateur père et la transmission du *beacon* qu'il doit émettre pour respecter la cascade. Par conséquent, la durée d'activation du module radio pour un coordinateur, notée δ_c, par rapport à celle d'une feuille est majorée par la durée de transmission d'un *beacon* d_{tb}, nous avons :

$$\delta_c[T_0; T_1] = \frac{d_{[T_0;T_1]}}{2} + d_{tb}.$$

Période $[T_1; T_2]$: La consommation énergétique d'un nœud durant la période $[T_1; T_2]$ dépend de la durée de la période d'activité d'une étoile. On considère que le module radio d'une feuille est actif durant au plus toute la période intra-étoile. Pour cette période, la durée pendant laquelle le module radio est actif pour une feuille, notée δ_f, est définie comme suit :

$$\delta_f[T_0; T_1] \leq d_{intra}.$$

La durée pendant laquelle le module radio est actif pour un coordinateur correspond à la durée cumulée de la période intra-étoile et n fois la durée de l'intervalle de relais (avec n étant le nombre de coordinateurs fils du nœud + 1). Nous avons donc :

Tableau 4.2 – Consommation énergétique d'une carte TelosB selon l'état du module radio.

État du module radio	**Consommation énergétique**
En sommeil	3 mW
En activité (réception / transmission à 0 dBm)	41 mW

$$\delta_c[T_0; T_1] = d_{intra} + n \times d_{relais}.$$

Période $[T_2; T_3]$ **:** Seuls les coordinateurs sont actifs durant toute la période $[T_2; T_3]$. La durée d'activation du module radio pour un coordinateur est définie comme suit :

$$\delta_c[T_0; T_1] = d_{[T_2;T_3]}.$$

Période $[T_3; T_0]$ **:** Durant cette période tous les nœuds du réseau place leur module radio en sommeil.

Cycle MaCARI : Lors d'un cycle MaCARI, la durée moyenne pendant laquelle une feuille active son module radio, notée Δ_f, est définie comme suit :

$$\Delta_f = \frac{d_{[T_0;T_1]}}{2} + d_{intra}.$$

Lors d'un cycle MaCARI, la durée moyenne pendant laquelle un coordinateur active son module radio, notée Δ_c (avec n le nombre de fils coordinateur du nœud +1) est calculée comme suit :

$$\Delta_c = \frac{d_{[T_0;T_1]}}{2} + d_{tb} + d_{intra} + n \times d_{relais} + d_{[T_2;T_3]}.$$

Les évaluations de consommation énergétique d'une carte TelosB dans [29] permettent d'établir l'impact de l'état du module radio CC2420 sur la consommation. Ces informations sont résumées dans le tableau 4.2.

L'utilisation du module radio est la principale source de consommation d'énergie : une carte TelosB consomme environ 14 fois plus lorsque le module radio est en activité que lorsqu'il est en sommeil. Pour l'implémentation, nous pouvons donc disposer d'une consommation énergétique selon le type de nœud $C_{nœud}$, exprimée en mW (avec CA

la consommation quand le module radio est actif et CS quand le module radio est en sommeil) telle que :

$$C_{noeud} = \frac{CA \times \Delta_{noeud} + CS \times (d_{cycle} - \Delta_{noeud})}{d_{cyle}}.$$

Chapitre 5

Mise en œuvre

Ce chapitre décrit la mise en œuvre de l'implémentation de MaCARI sur TelosB, pour une application permettant le contrôle de la ventilation dans une mine souterraine. Tout d'abord, nous présentons le contexte dans lequel la mise en œuvre de l'application a été réalisée. Puis, nous détaillons les besoins qui nous ont été décrits, les principes de l'application et son implémentation au dessus de la couche MaCARI. Ensuite, nous définissons la configuration de MaCARI permettant de répondre aux besoins applicatifs. Nous décrivons les mesures de portée et les déploiements des cartes TelosB sur le site. Enfin, nous proposons une évaluation du comportement de la solution dans la mine souterraine.

5.1 Contexte

La mise en œuvre d'une application destinée à l'industrie minière à été réalisée au sein d'un laboratoire de recherche à Val-d'Or, au Canada. Tout d'abord, nous décrivons le laboratoire de recherche Télébec en communication souterraines et le projet de recherche dans lequel cette partie de mon stage a été réalisée. Nous présentons ensuite la mine expérimentale CANMET, le site d'expérimentation de l'application. Enfin, nous présentons certaines spécificités des mines souterraines et une partie des enjeux de l'in-

dustrie minière.

5.1.1 Le laboratoire de recherche Télébec en communication souterraines

Cette phase de réalisation de mon stage s'est déroulée pendant deux mois au sein du Laboratoire de Recherche Télébec en Communication Souterraines (LRTCS [39]) de l'Université Québécoise d'Abitibi-Témiscamingue (UQAT) de Val-d'Or. Ce laboratoire collabore avec l'industrie minière à travers plusieurs projets de recherche visant à proposer de nouvelles solutions liées aux télécommunications sans fil dans les mines souterraines. Notamment, le projet de Démonstration Appliquée et Prototypage d'Outils Industriels (DAPOI) a pour but de définir les besoins pratiques et les défis à relever à court terme pour l'industrie minière. Il cible le déploiement de systèmes commerciaux adaptés au secteur minier.

Mon travail sur la mise en œuvre de l'application permettant le contrôle de la ventilation d'une mine, basée sur l'implémentation de MaCARI sur cartes TelosB, se situe dans le cadre du projet DAPOI. Cette tâche a été supervisée par N. Hakem, professeur-chercheur à l'Université du Québec en Abitibi-Témiscamingue et encadrée par mon tuteur de projet M. Ailas, ingénieur au LRTCS.

5.1.2 La mine-laboratoire CANMET

Le LRTCS dispose d'un accès aux installations de la mine-laboratoire, propriété de l'organisation CANMET [40] *(Canada Centre for Mineral and Energy Technology)*. La mine laboratoire CANMET, située à 14 km de Val-d'Or, est une ancienne mine d'or et de zinc aujourd'hui entretenue pour accueillir des expérimentations scientifiques. La mine est un site d'essai souterrain en roche dure qui comporte une rampe d'accès et cinq niveaux souterrains comptant environ 2500 mètres de galeries. La figure 5.1 présente de façon schématique une coupe de la mine où sont illustrés la rampe d'accès et les cinq niveaux souterrains (les trois niveaux disponibles pour les expérimentations sont ceux situés à 40, 50 et 70 mètres sous terre).

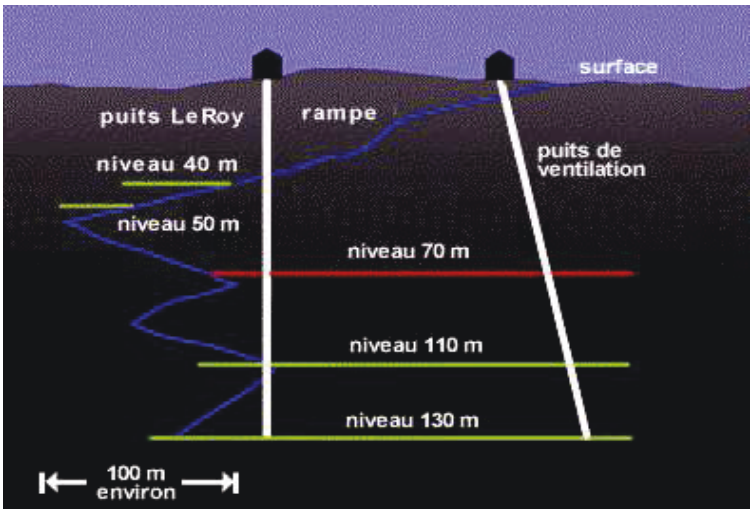


Figure 5.1 – Coupe transversale de la mine-laboratoire CANMET.

5.1.3 Les mines souterraines et l'industrie minière

Les mines souterraines ont des caractéristiques particulières que nous allons détailler brièvement. Rappelons qu'une mine souterraine est constituée d'un ensemble de galeries creusées dans le but d'exploiter un gisement de ressources. L'architecture d'une mine varie selon la nature du sol, de son relief et des ressources exploitées. Elle comporte les installations essentielles de soutènement, d'excavation des minerais et de ventilation de l'air. De plus, pour accroître la productivité et la sécurité, des réseaux d'électricité, d'éclairage, de détection et de télécommunication sont implantés de manière plus ou moins étendue selon la faisabilité. La figure 5.2 montre une photo d'une galerie de la mine-laboratoire CANMET.

Les principaux défis de l'industrie minière sont liés aux caractéristiques et contraintes du milieu souterrain. Les contraintes sont amplifiées par le phénomène d'épuisement des gisements qui impose une exploitation de ressources de plus en plus difficiles d'accès, augmentant proportionnellement les coûts en termes de prospection, d'équipements, d'infrastructures et de temps. Parallèlement, il est nécessaire de minimiser les risques supportés par le mineur (appelé aussi minier au Québec).



Figure 5.2 – Galerie d'une mine d'or à Val d'or (Canada).

5.2 Description de l'application et de son implémentation

Nous allons tout d'abord présenter la description de l'application de contrôle de la ventilation d'une mine et son intérêt pour la sécurité des travailleurs. Ensuite, nous décrivons l'implémentation de cette application au dessus de la couche MaCARI.

5.2.1 Description de l'application

La ventilation dans une mine souterraine est primordiale pour la sécurité de l'exploitation et des travailleurs. En effet, un air non brassé peut entraîner la présence stagnante de poussière, de gaz nocifs voire explosifs. Cependant, la ventilation constante dans une mine est très couteuse. Pour réduire son coût, le volume de la ventilation (i.e. le débit d'air) doit être contrôlé et adapté dans chaque partie de la mine, selon les besoins. Par exemple, lors de la détection d'un gaz particulier, de la présence d'un véhicule ou d'un équipement générant des gaz d'échappement souillant l'air, il faut augmenter le volume de la ventilation pour assainir l'air dans un délai très bref. Cette tâche est généralement réalisée par un système de surveillance centralisé et automatisé qui adapte le débit de l'air en fonction d'indications et de mesures relevées tout au long des galeries. Or le câblage nécessaire à cette surveillance est couteux et source de problèmes (pose et entretien dif-

ficile, encombrement, arrachages fréquents, etc).

L'application envisagée est basée sur l'utilisation d'un réseau de capteurs sans fil pour collecter les mesures et les transmettre au système de surveillance centralisé. Le réseau de capteurs sans fil remonte périodiquement des mesures prises en certains points stratégiques des galeries souterraines vers le nœud passerelle (relié au système de ventilation). De plus, chaque capteur est capable de générer une alarme pour déclencher une augmentation de la ventilation dans un temps restreint en cas d'urgence (suite à la détection de gaz nocif par exemple). Régulièrement, des trames de contrôle issues des extrémités de l'arbre sont envoyées pour tester l'intégrité du réseau. Cette application de ventilation nécessite une réactivité inférieure à 2 secondes pour des raisons de sécurité. La ventilation est déclenchée si une alarme atteint le nœud passerelle ou si le nœud passerelle ne reçoit pas de trames de contrôle pendant un certain temps (dont la durée doit être paramétrable).

5.2.2 Implémentation de l'application au dessus du protocole MaCARI

Cette utilisation de MaCARI se situe dans le domaine d'application de collecte de données environnementales, avec une collecte d'informations à destination d'un point centralisé (que nous placerons à la racine de l'arbre du réseau, c'est-à-dire au CPAN). Nous décrivons dans cette partie comment est implémentée l'application de contrôle de ventilation au dessus du protocole MaCARI.

Initialement, chaque étoile de l'arbre est composée d'un coordinateur et de n feuilles (n étant compris entre 0 et 8). Chaque feuille émet une trame d'entretien (en trafic non prioritaire) tous les m cycles pour signaler au coordinateur qu'elle est toujours opérationnelle : au bout d'un certain nombre (paramétrable) de cycles, si le coordinateur n'a pas reçu de trame d'entretien d'une feuille, le coordinateur remonte l'information au CPAN. De plus, les feuilles, munies de capteurs, surveillent en permanence la composition de l'air. Lorsque l'air est détecté souillé, la couche application de la feuille envoie une commande d'émission d'une trame alarme en direction du coordinateur et à destination du CPAN. L'alarme, estampillée temporellement par MaCARI lors de sa génération, est émise durant

la période d'activité intra-étoile (dans l'intervalle $[T_1; T_2]$ du cycle MaCARI). Deux cas peuvent donc se produire au niveau du coordinateur durant la période intra-étoile : soit il reçoit une ou plusieurs alarmes provenant des feuilles (dans ce cas, l'application place l'état de l'étoile en alarme), soit il ne reçoit pas d'alarme (dans ce cas, l'application place l'état de l'étoile en état normal).

Chaque branche de l'arbre reliée au CPAN remonte au CPAN une trame de contrôle par cycle en trafic prioritaire. Une trame de contrôle est générée par chaque coordinateur situé à l'extrémité de l'arbre puis remontée vers le CPAN. La charge utile d'une trame de contrôle varie selon la portée des nœuds et la zone à couvrir. L'état de chaque étoile du réseau y est ordonné et codé sur deux bits de la manière suivante :

- si l'état d'une étoile vaut 00, l'étoile n'a pas donné de réponse,
- si l'état d'une étoile vaut 01, l'étoile est en état normal (pas de problème détecté),
- si l'état d'une étoile vaut 10, l'étoile est en état normal mais au moins une de ses feuilles n'est pas opérationnelle,
- si l'état d'une étoile vaut 11 : l'étoile est en état d'alarme.

Lors de la création d'une trame de contrôle, tous les bits de la charge utile sont placés à 0. Cette trame est ensuite modifiée à chaque saut : le coordinateur de chaque étoile met à jour l'état de son étoile dans la charge utile, un champ lui étant réservé en fonction de sa place dans la branche de l'arbre. Lorsqu'un coordinateur reçoit plusieurs trames de contrôle, il fusionne les bitmaps de chacune des trames (avec un OU logique) puis transmet le résultat dans une trame de contrôle à son coordinateur père. Ainsi, à T_2, le CPAN connaît l'état de chaque étoile grâce aux informations contenues dans la charge utile de chaque trame de contrôle reçue. Si une étoile est en état d'alarme, l'application peut commander l'augmentation du volume de la ventilation localisée sur la position de l'étoile ayant signalé l'alarme. Si une étoile ou un groupe d'étoiles est dans l'état 00, ou si le CPAN ne reçoit pas de trame de contrôle d'une branche pendant un certain nombre de cycles, alors la ventilation est également déclenchée au niveau de toutes les étoiles concernées. Enfin, l'état 10 d'une étoile, signalant qu'au moins une de ses feuilles n'est plus opérationnelle, peut être utilisé pour faciliter les opérations de maintenance.

5.3 Configuration de MaCARI

La réactivité de l'application dépend de la durée d'un cycle MaCARI et du délai d'acheminement du trafic prioritaire en direction du CPAN. Dans le cadre de cette application où le trafic prioritaire est orienté vers le CPAN, nous configurons le séquencement des périodes d'activité durant $[T_1; T_2]$ de manière à favoriser le trafic montant (il n'y a donc pas d'alternance de l'ordre de séquencement). Nous pouvons alors considérer le délai d'acheminement de bout en bout du trafic prioritaire dans le pire cas possible : une feuille de la première étoile active lors de $[T_1; T_2]$ génère une alarme en fin de période d'échange intra-étoile, mais n'a pas assez de temps pour l'envoyer pendant le cycle actuel. Elle reporte l'envoi au prochain cycle. Dans ce cas, nous obtenons un délai de bout en bout maximum égale à un cycle plus une période $[T_1; T_2]$. Or, l'application nécessite une réactivité inférieure à 2 secondes ; le délai d'acheminement maximum est donc défini comme suit :

$$d_{[T_0;T_1]} + 2 \times d_{[T_1;T_2]} + d_{[T_2;T_3]} + d_{[T_3;T_0]} \leq 2.$$

La durée du cycle est liée au dimensionnement des périodes d'activité et au nombre d'étoiles dans le réseau. Nous avons utilisé les 12 cartes TelosB disponibles pour l'expérimentation dans la mine. Nous avons choisi une topologie linéaire afin de maximiser l'étendue du réseau et de présenter ainsi un pire des cas. Le réseau de l'expérimentation est composé de 10 coordinateurs (dont le CPAN) et d'une feuille (la carte restante étant utilisée comme *sniffer*), comme illustré sur la figure 5.3.

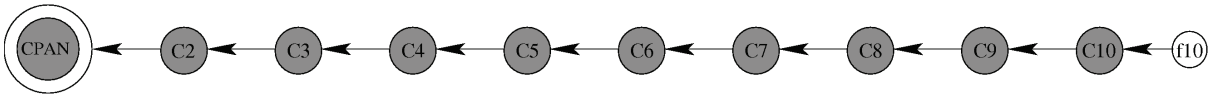


Figure 5.3 – Topologie du réseau linéaire utilisé pour l'expérimentation de l'application.

Pour dimensionner les périodes, nous utilisons les formules décrites en 4.3.1. Comme l'application a des exigences de réactivité forte, notre configuration vise à minimiser la durée de chacune des périodes. Pour les périodes $[T_1; T_3]$, l'unité de temps utilisée est le slot d'activité MaCARI : nous choisissons de fixer l'exposant de durée d'un slot (EDS) à 3.

$$d_{slot} = 2^3 \times 960 \times 10^{-3} = 7,68\ ms.$$

La durée de la période de synchronisation $[T_0; T_1]$ pour 10 coordinateurs, exprimée en ms, est définie par la formule suivante :

$$d_{[T_0;T_1]} = 10 \times (0,18 \times 10 + 2,8) = 46\ ms.$$

La durée de la période $[T_1; T_2]$ dépend de la durée de la période intra-étoile, de l'intervalle de relais et de la durée de la période de garde. Nous fixons le nombre de slots d'activité à 4 pour la période intra-étoile, à 2 pour l'intervalle de relais et une période de garde de $0,640$ ms. Nous pouvons donc déterminer les durées suivantes :

$$d_{intra} = 4 \times d_{slot} = 4 \times 7,68 = 30,72\ ms.$$

$$d_{relais} = 2 \times d_{slot} = 2 \times 7,68 = 15,36\ ms.$$

$$d_{etoile} = 30,72 + 15,36 + 2 \times 0,64 = 47,36\ ms.$$

$$d_{[T_1;T_2]} = 47,36 \times 10 = 473,6\ ms.$$

Nous considérons que pour avoir une durée de $[T_2; T_3]$ suffisante pour le routage de l'ensemble du trafic non prioritaire généré lors d'un cycle (l'étude précise du dimensionnement de cette période est hors sujet de ce mémoire), nous pouvons la fixer à la durée de $[T_1; T_2]$ sans les périodes de garde soit :

$$d_{[T_2;T_3]} = 473,6 - (2 \times 0,64 \times 10) = 460,8\ ms.$$

Nous fixons la durée de la période d'inactivité à une durée volontairement réduite telle que :

$$d_{[T_3;T_0]} = 30\ ms.$$

Avec les durées définies, nous pouvons donner la durée d'un cycle global de MaCARI pour cette topologie, soit :

$$d_{cycle} = d_{[T_0;T_1]} + d_{[T_1;T_2]} + d_{[T_2;T_3]} + d_{[T_3;T_0]} = 1010,4\ ms.$$

La durée maximale d'acheminement de bout en bout d'une trame émise en trafic prioritaire vaut donc :

$$d_{[T_0;T_1]} + 2 \times d_{[T_1;T_2]} + d_{[T_2;T_3]} + d_{[T_3;T_0]} = 1484\ ms.$$

On obtient une durée de cycle global MaCARI de $1,0104$ secondes avec un délai de bout en bout maximum de $1,484$ secondes, ce qui correspond aux exigences de l'application (délai inférieur à 2 secondes).

5.4 Étude de portée et déploiement dans la mine

L'étendue de la zone couverte est dépendante de la portée des cartes TelosB. Nous avons réalisé une étude de la portée des cartes TelosB en milieu souterrain afin de déterminer une manière de déployer les noeuds dans la mine CANMET.

5.4.1 Étude de la portée des cartes TelosB

L'objectif de cette étude est de mesurer l'impact de l'environnement sur la portée des cartes TelosB. Nous avons tout d'abord évalué cette portée en milieu ouvert, puis en milieu souterrain.

5.4.1.1 Portée en milieu ouvert sans obstacle

Le module radio CC2420 des cartes TelosB émet sur la plage de fréquence de $2,4$ GHz à $2,4835$ GHz. Pour évaluer la portée, nous faisons varier la distance entre un émetteur et un récepteur, puis nous mesurons la puissance du signal reçu (RSSI : *Received Signal Strength Indicator*, exprimé en dBm).

La figure 5.4 montre l'évolution du RSSI au fil de la distance. Chaque point de cette figure représente une moyenne de 20 mesures du RSSI en milieu ouvert sans obstacle, en augmentant la distance entre les dispositifs d'un pas de $8,5$ mètres environ, avec une puissance d'émission de 0 dBm.

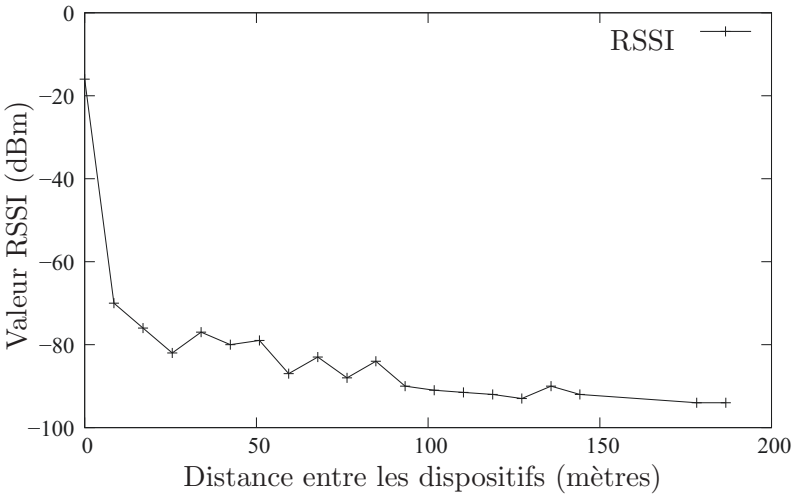


Figure 5.4 – Étude de la portée en milieu ouvert sans obstacle.

Comme le montre la figure 5.4, la portée maximum en conditions optimales (dispositifs en ligne de vue et en milieu ouvert) est d'environ 180 mètres. Cette mesure permet d'avoir un point de comparaison par rapport aux mesures relevées dans la mine.

Comme vu en 2.1.1.1, le module radio CC2420 fournit l'indicateur de qualité de lien LQI, mesurant le taux d'erreur binaire dans une trame reçue. Pour rappel, le LQI a pour but de donner une estimation le taux de pertes de trames entre deux entités du réseau.

5.4.1.2 Portée en milieu souterrain

Les mesures de portée de cette partie sont réalisées au niveau -70 mètres de la mine-laboratoire CANMET, composé de galeries étroites ($2,5 \times 2,5$ mètres en moyenne). Pour les résultats suivants, chaque point représente une moyenne de 10 mesures relevées dans des portions de la mine pour plusieurs contextes. Les cartes sont placées à exactement $2,20$ mètres du sol au centre de la galerie. Nous faisons varier la distance entre les dispositifs d'un pas de 5 mètres en déplaçant le récepteur (placé sur une perche de $2,20$ mètres). Pour chaque expérimentation, l'émetteur envoie son signal avec une puissance de transmission de 0 dBm. Nous mesurons le RSSI, le LQI et le nombre de trames reçues dans le but d'étudier une corrélation entre ces résultats.

La figure 5.5 représente la disposition dans la mine de l'expérience dans un contexte de portée en ligne de vue (LOS : *Line Of Sight*) dans une galerie de 110 mètres en ligne droite.

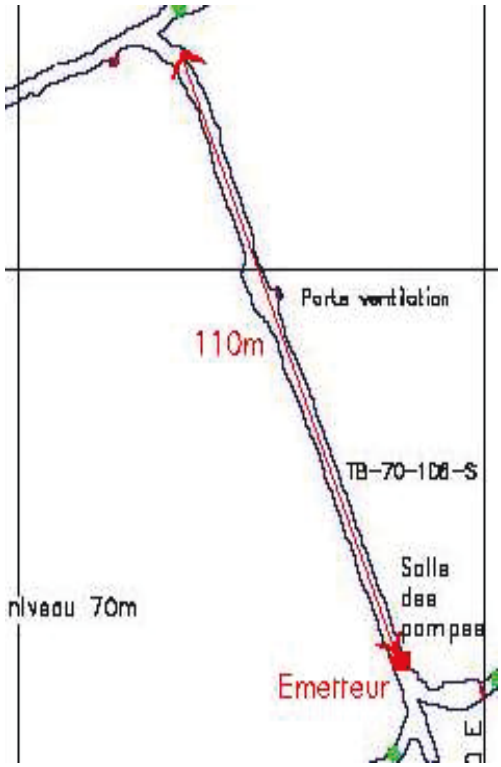
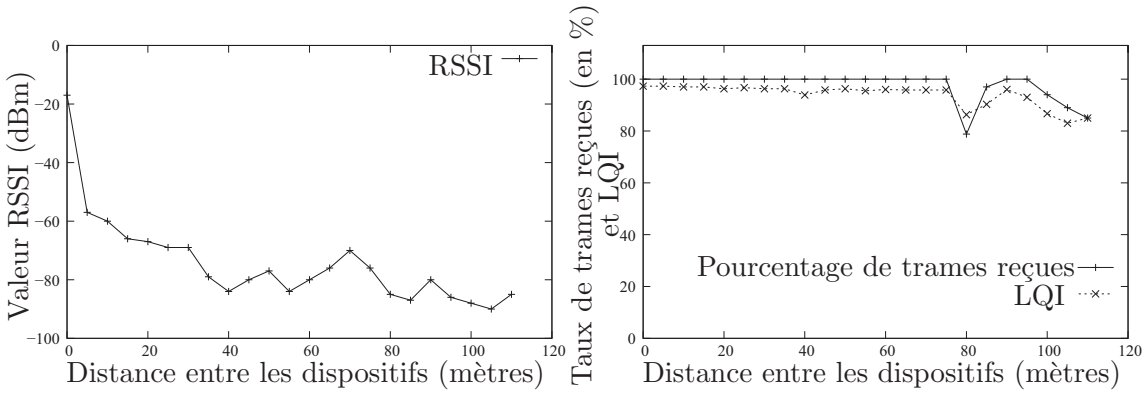


Figure 5.5 – Disposition dans la mine de l'expérimentation, en LOS.



(a) (b)

Figure 5.6 – Mesures du RSSI, du taux de trames reçues et du LQI, en LOS.

La figure 5.6 montre l'évolution du RSSI (à gauche figure 5.6(a)) et la corrélation entre le LQI et le taux de trames reçues (à droite figure 5.6(b)) en fonction de la distance entre les dispositifs. Les mesures décrites dans cette figure 5.6 sont réalisées en condition favorisant la portée (LOS). Nous observons une dégradation du taux de trames reçues autour de 100 mètres, quand le RSSI atteint −90 dBm. Le LQI moyen semble apporter une bonne corrélation avec le taux de trames reçues.

La figure 5.7 représente la disposition dans la mine de l'expérience dans un contexte de portée en non ligne de vue *(NLOS : Not Line Of Sight)* dans un virage de la rampe.

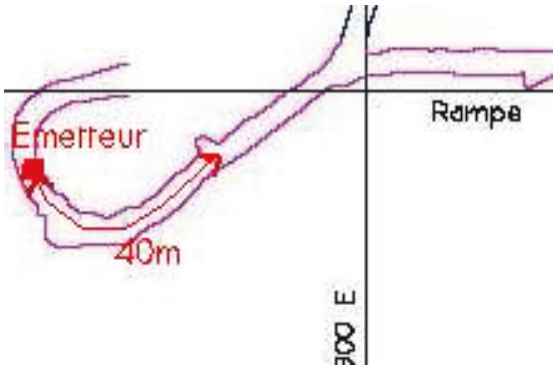


Figure 5.7 – Étude de la portée dans un virage de la rampe, en NLOS.

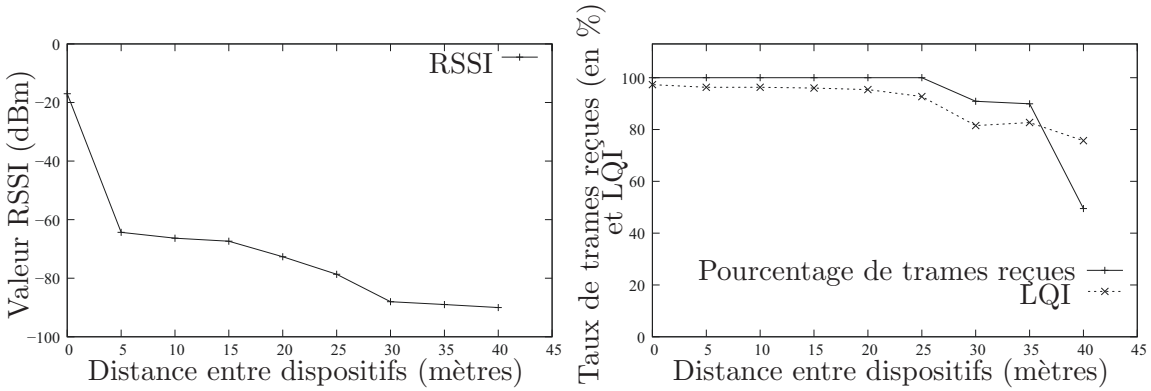


Figure 5.8 – Mesures du RSSI, du taux de trames reçues et du LQI, en NLOS.

La figure 5.8 montre l'évolution du RSSI (à gauche figure 5.8(a)) et la corrélation entre le LQI et le taux de trames reçues (à droite figure 5.8(b)) en fonction de la distance entre les dispositifs dans des conditions NLOS. Nous observons, à partir des relevés montrés sur cette figure, l'influence de l'effet de guide d'onde de la rampe qui permet la réception du signal en NLOS par un phénomène de guidage du signal le long de la galerie. Nous

observons une dégradation franche du signal dès 40 mètres. Nous notons que le LQI donne une bonne indication du taux de pertes de trames dans ces conditions.

La figure 5.9 représente la disposition dans la mine de l'expérience dans un contexte de portée en NLOS à un carrefour au relief très inégal.

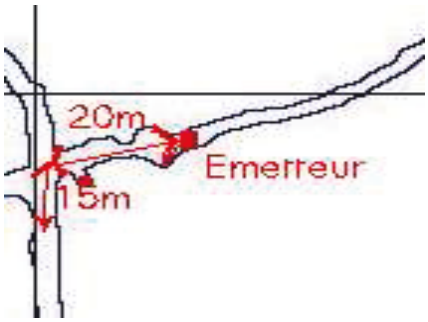


Figure 5.9 – Étude de la portée dans le cas d'un relief inégal, en NLOS.

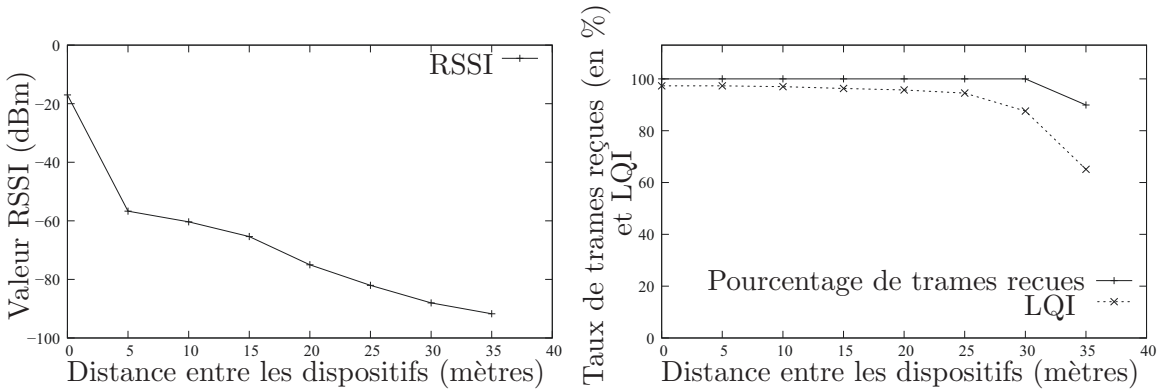


Figure 5.10 – Mesures du RSSI, du taux de trames reçues et du LQI, pour un relief inégal, en NLOS.

La figure 5.10 montre l'évolution du RSSI (à gauche figure 5.10(a)) et la corrélation entre le LQI et le taux de trames reçues (à droite figure 5.10(b)) en fonction de la distance entre les dispositifs dans des conditions NLOS. Les mesures décrites dans la cette figure ont été relevées en condition NLOS à partir de 20 mètres avec un effet guide d'onde très réduit du fait du relief inégal. Nous constatons que le taux de trames reçues chute dès 30 mètres. Nous constatons également que l'observation du LQI permet une bonne corrélation avec le taux de trames reçues dans ces conditions.

5.4.1.3 Conclusion sur la portée des cartes TelosB en milieu souterrain

La portée d'une carte TelosB est impactée par les reliefs et la disposition des galeries. D'après les mesures effectuées, nous pouvons donner une fourchette concernant le nombre de nœuds nécessaires pour couvrir un kilomètre de galerie selon le type de cette galerie. Ainsi pour une puissance d'émission à 0 dBm, la densité des nœuds est de l'ordre de 32 coordinateurs par kilomètre dans le cas de conditions difficile (NLOS sans guide d'ondes) et de 8 coordinateurs par kilomètre (LOS). De plus, nous observons que le LQI est un indicateur représentatif du taux de trames reçues dans ce type de milieu.

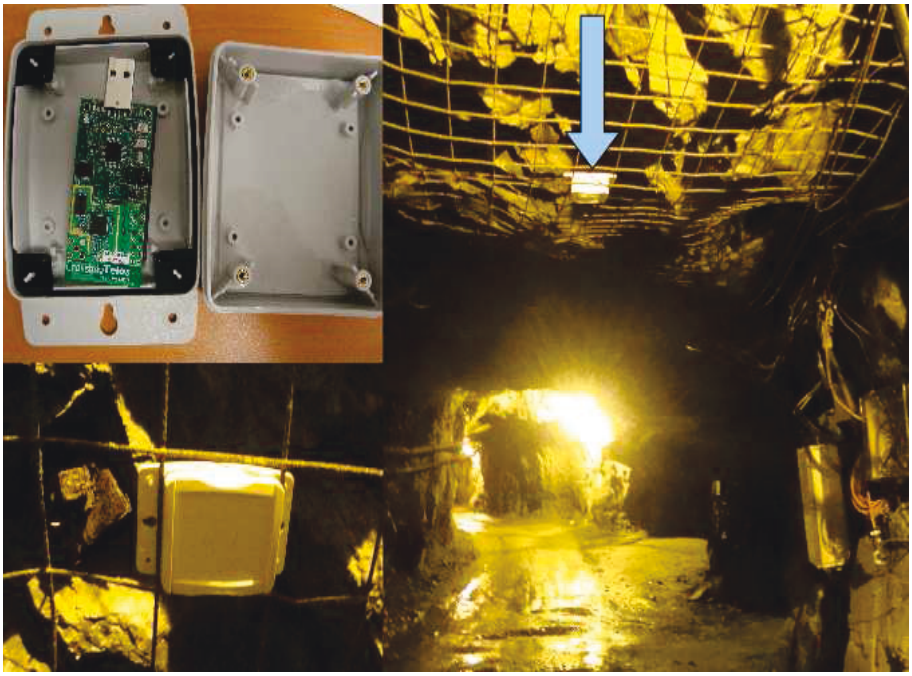


Figure 5.11 – Boitier étanche contenant une carte TelosB (en haut à gauche), et photos de la disposition du CPAN dans la mine.

5.4.2 Déploiements dans la mine

Pour expérimenter l'application de contrôle de la ventilation, dix cartes TelosB ont été réparties dans les galeries de façon linéaire dans le sous niveau situé à 70 mètres sous terre. Elles ont été configurée pour une puissance d'émission de 0 dBm. Pour rappel, le choix d'une topologie linéaire a été effectué pour pouvoir maximiser l'étendue du réseau et sa profondeur. Les cartes TelosB ont été placées dans des boîtes étanches fixées sur la voute des galeries. La figure 5.11 est un assemblage de trois photos détaillant le type

de boîte étanche utilisée et la disposition du CPAN sur une grille de soutènement de la mine.

Deux déploiements ont été réalisés. Pour le déploiement A, les nœuds ont été disposés comme illustré sur la figure 5.12. La distance entre le CPAN et le coordinateur C_{10} est d'environ 550 mètres, ce qui correspond à une densité moyenne de 18 coordinateurs par kilomètre. La zone de couverture recouvre environ 700 mètres de galeries.

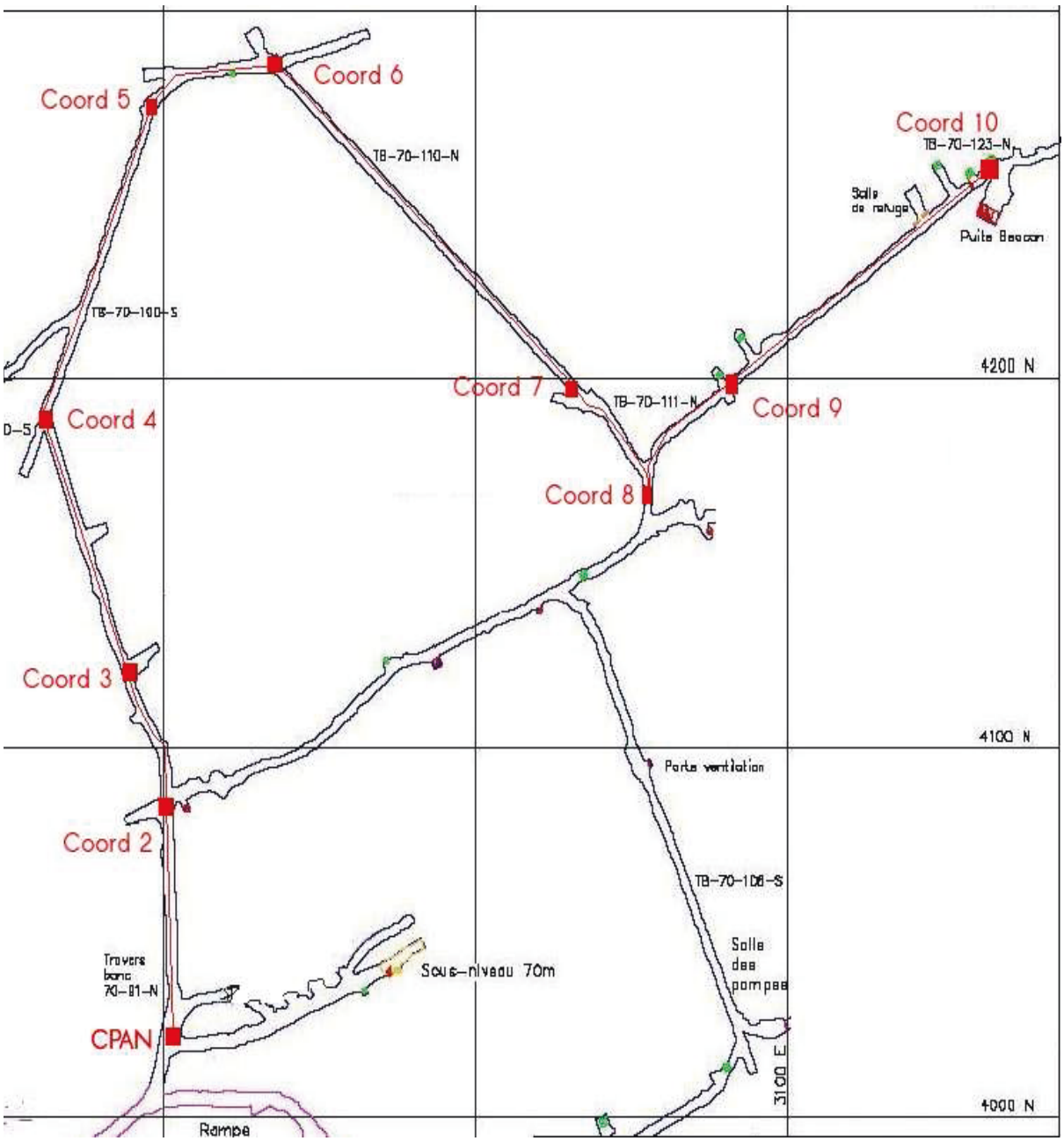


Figure 5.12 – Disposition des nœuds pour le déploiement A dans la mine.

Pour le déploiement B, les nœuds ont été disposés comme illustré sur la figure 5.13. La distance entre le CPAN et le coordinateur 10 est d'environ 500 mètres, soit une densité moyenne de 20 coordinateurs par kilomètre. La zone de couverture recouvre environ 650 mètres de galeries.

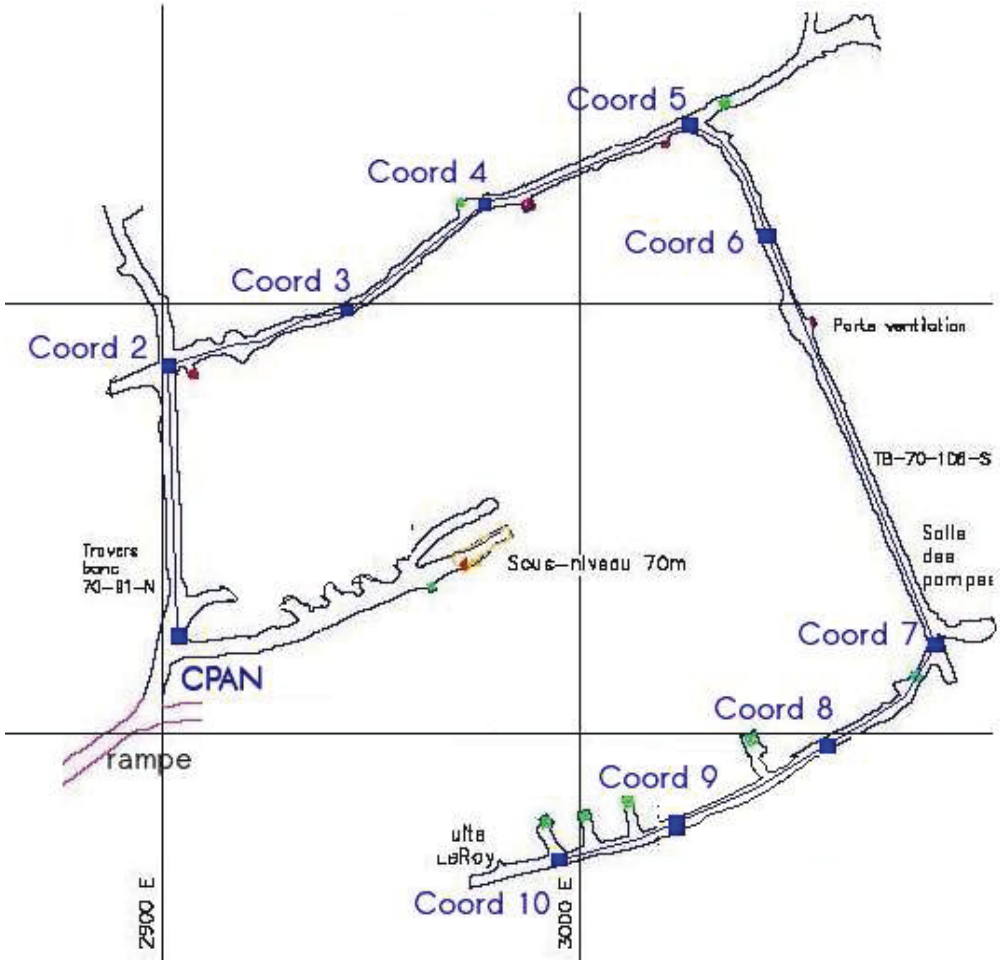


Figure 5.13 – Disposition des nœuds pour le déploiement B dans la mine.

5.5 Évaluation de l'application

L'objectif est d'abord de générer un trafic non prioritaire et d'expérimenter son comportement, puis d'évaluer les conditions réservées au trafic prioritaire, et enfin de donner une estimation de la consommation énergétique selon le type de nœuds composant le réseau.

5.5.1 Étude du trafic non prioritaire

Le trafic non prioritaire est routé pendant la période $[T_2; T_3]$ en utilisant le routage hiérarchique. Comme nous l'avons identifié dans 4.2.3.3, dans le cas où le module radio est placé en sommeil durant les *backoffs*, la réalisation des relais de trames en CSMA/CA slotté est fortement perturbée pour ce type de configuration. Pour éviter ce phénomène, nous avons choisi de générer un trafic de type non prioritaire à partir du coordinateur C_{10} (le plus éloigné du CPAN) en introduisant un mécanisme permettant d'assurer un relais de trame fonctionnel : à T_2, le coordinateur C_{10} génère un nombre n de trames de 30 octets dont la charge utile contient le nombre de trames que le coordinateur possède dans sa file d'attente. Le coordinateur père reste donc en écoute et ne relaye pas de trames tant qu'il n'a pas reçu les n trames de son coordinateur fils. Les trames sont émises sans retransmission pour évaluer le taux de pertes de trames selon le volume de trafic généré pour une durée $[T_2; T_3]$ fixe.

Dans un premier temps, nous étudions le taux de trames reçues par le CPAN issues de l'étoile du coordinateur C_{10} (pour les déploiements A et B) pendant la période $[T_2; T_3]$. Pour rappel, la période $[T_2; T_3]$ a une durée de $460,8$ ms. La topologie linéaire implique de nombreux relais de trames : chacune des trames émises par le coordinateur C_{10} doit être relayée par les huit autres coordinateurs pour atteindre le CPAN.

Dans un premier temps, le taux de trames reçues par le CPAN est évalué en faisant varier le nombre de trames émises par le coordinateur C_{10}. Ce taux représente le nombre de trames reçues par le CPAN par rapport au nombre de trames émises par C_{10} pour chaque cycle (sur la figure 5.14), il est exprimé en pourcentage sur l'axe vertical de gauche). Le nombre de transmissions de trames induit par le nombre de sauts entre l'émetteur et le récepteur constitue le volume total du trafic ; il est décrit sous forme d'histogrammes et exprimé en nombre de trames sur l'axe vertical de droite sur la figure 5.14. Chaque point représente une moyenne des mesures pour 10 répétitions de l'expérience. Tous les nœuds émettent à une puissance de 0 dBm.

La figure 5.14 montre que la durée de la période $[T_2; T_3]$ est suffisante pour absorber le trafic associé à la génération de 5 trames par le coordinateur C_{10} (se trouvant à 9 sauts), soit un total de 45 trames circulant par cycle. Nous observons une dégradation

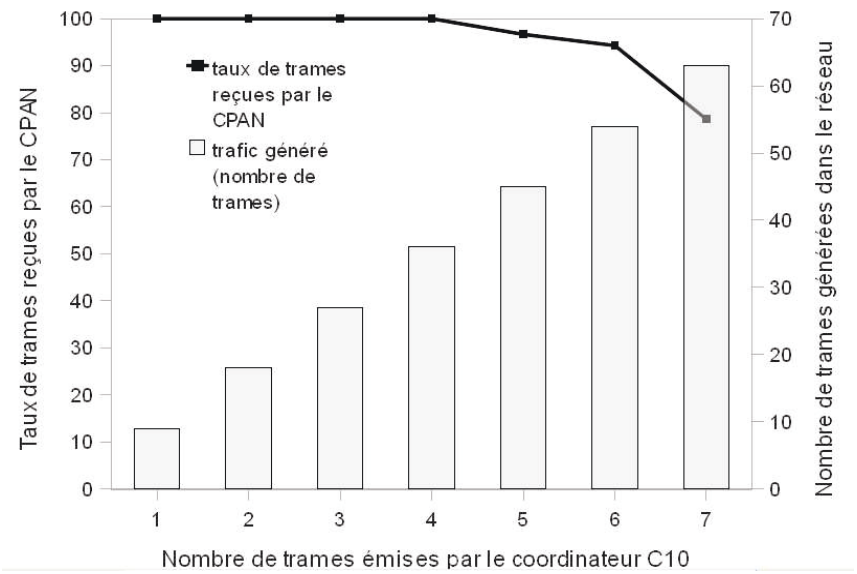


Figure 5.14 – Étude du trafic non prioritaire généré par le coordinateur C_{10} durant $[T_2; T_3]$.

des performances pour un nombre de trames générées supérieur.

Nous étudions ensuite le comportement du trafic durant la période $[T_2; T_3]$, quand les trames sont générées par les coordinateurs C_4, C_7 et C_{10} qui sont hors de portée les uns des autres avec le même protocole expérimental. Les nœuds étant géographiquement éloignés, nous pouvons émettre l'hypothèse que certaines transmissions puissent avoir lieu parallèlement.

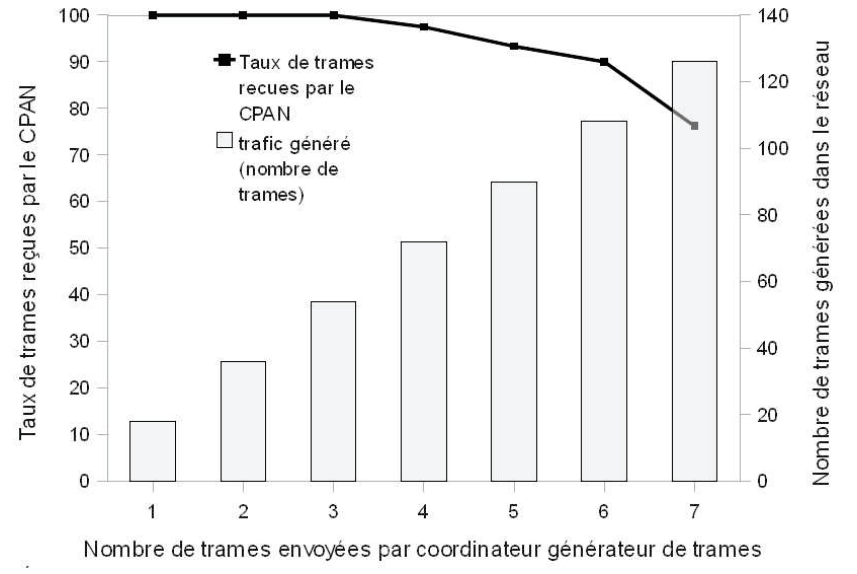


Figure 5.15 – Étude du trafic non prioritaire généré par trois coordinateurs géographiquement éloignés durant $[T_2; T_3]$.

La figure 5.15 montre l'évolution du taux de trames reçues par le CPAN et le nombre de trames générées dans le réseau par trois coordinateurs isolés les uns des autres. Sur cette figure, nous observons que le taux de trames reçues par le CPAN pour 6 trames envoyées par nœud reste supérieur à 90% alors que le trafic total dans le réseau est 50% plus important que le cas décrit sur la figure 5.14. Nous pouvons donc conclure qu'il y a un phénomène d'envois simultanés de trames qui influe positivement sur la capacité du trafic. Dans cette topologie linéaire, l'éloignement entre les nœuds du réseau favorise le trafic non contraint en période de routage.

5.5.2 Évaluation du délai de bout en bout du mécanisme d'alarme

Cette application de surveillance de la ventilation dans une mine peut mettre en évidence les avantages de la différenciation de service fournie par MaCARI. En effet, quelque soit l'intensité du trafic non garanti, le trafic garanti ne devrait pas être impacté. Pour le prouver, nous avons évalué l'application de contrôle de la ventilation en la faisant cohabiter avec une autre application émettant un trafic non prioritaire. Pour ce faire, nous avons décider que le coordinateur C_{10} enverrait 5 trames de 30 octets de type non prioritaire à chaque cycle. Ce trafic non prioritaire pourrait, par exemple, véhiculer des relevés de mesures environnementales.

Pour vérifier la bonne marche du mécanisme d'alarme, il faut se placer dans le pire des cas. Pour cela, la feuille de l'étoile la plus profonde de l'arbre est choisie comme source de l'alarme. Ainsi, le coordinateur C_{10} (pour les déploiements A et B) sera le nœud à l'initiative du routage des alarmes. Lors de chaque cycle, une trame de contrôle sera donc remontée du coordinateur C_{10} au CPAN en trafic prioritaire : elle signalera la présence (ou l'absence) d'une étoile en état d'alarme.

Pour évaluer la stabilité de l'application, nous avons réalisé une évaluation du mécanisme pendant une heure environ. Au cours de cette durée, 60 alarmes sont générées par la feuille du coordinateur C_{10} à des instants choisis aléatoirement chaque minute. La charge utile d'une trame d'alarme contient l'ensemble des états de chacune des étoiles (codé sur 3 octets), le numéro de séquence du cycle pendant lequel elle a été générée (codé sur 2 octets) et l'instant de génération de l'alarme (codé sur 4 octets). Nous évaluons le temps entre

Tableau 5.1 – Résultats de l'expérimentation du mécanisme d'alarme.

Nombre de cycles	3876
Nombre de trames de contrôle reçues par le CPAN	3876
Nombre d'alarmes envoyées	60
Nombre d'alarmes reçues par le CPAN	60
Délai d'acheminement minimum	0,493 s
Délai d'acheminement moyen	0,987 s
Délai d'acheminement maximum	1,476 s
Borne théorique calculée en 5.3	1,484 s
écart type	0,495 s

l'émission de l'alarme par la feuille et son arrivée au CPAN en utilisant l'estampillage temporel fourni par MaCARI. Les résultats de cette expérimentation sont illustrés dans le tableau 5.1 et la figure 5.16.

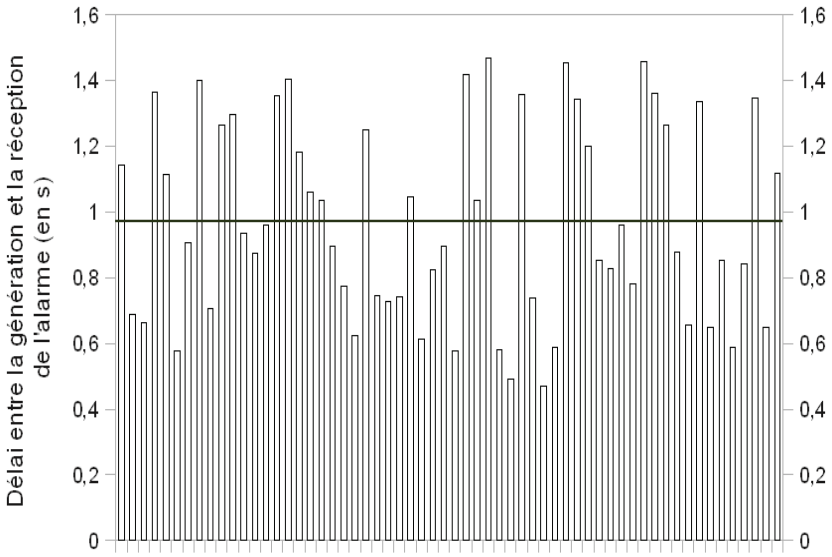


Figure 5.16 – Représentation des délais d'acheminement mesurés entre l'émission et la réception de chaque alarme émise.

La figure 5.16 montre le délai, pour chacune des 60 alarmes, entre la génération par la feuille de C_{10} et la réception par le CPAN. À partir de cette figure, nous observons que le délai de bout en bout est de $0,987$ secondes en moyenne, avec un délai maximum de $1,476$ secondes, ce qui est en accord avec les besoins de l'application de contrôle de la ventilation. De plus, le tableau 5.1 montre que le réseau est stable (i.e. aucune trame de contrôle n'a été perdue).

5.5.3 Consommation énergétique

La consommation énergétique de l'application est variable selon le type de chacun des nœuds. L'application nécessite une configuration de MaCARI qui maximise la réactivité : nous nous trouvons dans le cas où l'économie d'énergie apportée par MaCARI est la moins importante. Nous basons notre estimation des consommations sur les formules déterminées en 4.4 en faisant l'hypothèse que la consommation est supposée proportionnelle au temps de fonctionnement dans deux états caractéristiques (module radio actif et inactif). Pour la configuration établie en 5.3, la durée Δ_f pendant laquelle le module radio est actif pour une feuille est telle que :

$$\Delta_f = \frac{d_{[T_0;T_1]}}{2} + d_{intra} = 53,72\ ms.$$

La durée Δ_c pendant laquelle le module radio est actif pour un coordinateur est telle que :

$$\Delta_c = \frac{d_{[T_0;T_1]}}{2} + d_{beacon} + d_{intra} + n \times d_{relais} + d_{[T_2;T_3]} = 532,88\ ms.$$

Or, un cycle dure $1,0104$ secondes. Nous pouvons donc déduire que pour cette implémentation de l'application, le module radio est actif pendant $3,32\%$ d'un cycle pour une feuille et pendant $52,74\%$ pour un coordinateur. Pour rappel la consommation d'une carte TelosB est de 3 mW lorsque le module radio est en sommeil, et de 41 mW lorsqu'il est actif.

La consommation énergétique moyenne pour une feuille, notée C_f, et exprimée en mW est donc :

$$C_f = \frac{(41 \times \Delta_f + 3 \times (d_{cycle} - \Delta_f))}{d_{cycle}} = 5,02\ mW.$$

La consommation énergétique moyenne pour une coordinateur, notée C_c, et exprimée en mW est donc :

$$C_c = \frac{(41 \times \Delta_c + 3 \times (d_{cycle} - \Delta_c))}{d_{cycle}} = 23,04\ mW.$$

La consommation énergétique moyenne d'un coordinateur est donc environ $4,6$ fois

supérieure à celle d'une feuille. Par conséquent, si tous les nœuds sont équipés de batteries similaires, les coordinateurs épuiseront leur batterie plus rapidement : la durée de vie du réseau est bornée par l'autonomie des coordinateurs.

5.5.3.1 Bilan

L'évaluation de la solution dans la mine souterraine CANMET montre que l'application implémentée sur la couche MaCARI est fiable et peut supporter la présence d'un trafic additionnel de type non prioritaire. Le délai d'acheminement de bout en bout d'une alarme varie uniformément entre $0,493$ et $1,476$ secondes, ce qui répond aux exigences de latence de l'application. La consommation énergétique moyenne est de $5,02$ mW pour une feuille et de $23,04$ mW pour un coordinateur.

Chapitre 6

Conclusion

Les réseaux de capteurs sans fil doivent répondre aux besoins applicatifs des industriels : selon le profil de l'application, ces réseaux doivent fournir les fonctionnalités nécessaires et s'adapter à des exigences de latence, de qualité de service et/ou d'autonomie. Le projet OCARI propose une pile protocolaire pour les réseaux de capteurs sans fil utilisant la méthode d'accès MaCARI, qui permet de répondre à certains besoins applicatifs industriels, notamment en termes d'économie d'énergie et de déterminisme. Une implémentation de MaCARI existe sur une plateforme matérielle propriétaire impliquant des limitations quant à la propriété intellectuelle du code.

Ce mémoire décrit d'abord la réalisation de l'adaptation et du portage de MaCARI sur une plateforme *open-source*. Puis, il détaille le développement et la mise en œuvre d'une application basée sur cette implémentation destinée à l'industrie minière et testée dans une mine-laboratoire canadienne.

Dans le cadre de ce mémoire, nous avons tout d'abord décrit les principaux standards dédiés aux réseaux de capteurs sans fil. Chacun de ces standards possède des caractéristiques propres qui peuvent constituer des limites incompatibles avec les demandes des industriels. Nous avons alors détaillé la méthode d'accès MaCARI qui permet de répondre de manière flexible à ces demandes selon les exigences applicatives, en termes de latence, de différenciation de services, de déterminisme et d'autonomie énergétique. Nous

avons également présenté les caractéristiques de la carte TelosB et le système d'exploitation TinyOS, une plateforme complète entièrement *open-source*, constituant le support de l'implémentation de MaCARI.

6.1 Contributions

L'adaptation et la programmation des principes de MaCARI sur la plateforme TelosB a nécessité la conception d'une architecture basée sur l'implémentation du standard IEEE 802.15.4 et issue de l'Université Technique de Berlin. La version de MaCARI sur cartes TelosB est organisée en composants logiciels imbriqués pouvant disposer d'un accès exclusif à la ressource radio *via* un mécanisme d'arbitrage par attribution de jeton. Ces composants assurent les fonctionnalités de la couche protocolaire ainsi que le séquencement de la période de synchronisation et des différentes périodes du cycle MaCARI. Suite à ce développement, nous avons ajusté les paramètres propres à la plateforme TelosB dans l'implémentation afin d'optimiser les performances de transmission et le dimensionnement dynamique de la période de synchronisation.

Nous avons ensuite évalué l'implémentation à travers quatre points clés. Tout d'abord, nous avons évalué la qualité de la synchronisation. Dans un réseau MaCARI, toutes les entités sont synchronisées par le coordinateur du PAN. La précision de cette synchronisation est essentielle au bon fonctionnement des principes de MaCARI. Nous avons relevé une qualité de précision inférieure à 200 μs pour un réseau d'une dizaine de nœuds. Ensuite, nous avons évalué les méthodes d'accès utilisées dans MaCARI, notamment le CSMA/CA slotté issu du standard IEEE 802.15.4. L'évaluation des performances de la période intra-étoile durant $[T_1; T_2]$ (pour une topologie en étoile avec un nombre varia-ble de feuilles associées à chaque coordinateur) donne des résultats comparables à ceux simulés ou mesurés sur d'autres plateformes par le LATTIS et par le LIMOS. L'évaluation de la période de routage $[T_2; T_3]$ a révélé la nécessité d'une réflexion sur le comportement du CSMA/CA slotté durant les *backoffs* : le taux de pertes de trames varie selon que la réception de trames soit autorisée ou non durant les *backoffs*. En d'autres termes, il faut déterminer si le CSMA/CA slotté utilisé par les coordinateurs lors de la période $[T_2; T_3]$ favorise la transmission ou la réception des trames. Par la suite, nous avons déterminé analytique-

ment, pour cette implémentation de MaCARI sur cartes TelosB, les formules de calcul du délai d'acheminement de bout en bout, selon la destination du trafic prioritaire, ainsi que les formules permettant d'évaluer la consommation énergétique de chaque entité du réseau. Ces évaluations ont permis d'obtenir une vision d'ensemble de l'implémentation de MaCARI sur TelosB dont le comportement est représentatif et conforme aux comportements observés dans les simulations.

Enfin, nous avons développé une application permettant un contrôle de la ventilation dans une mine, basée sur l'implémentation de MaCARI sur cartes TelosB. L'application véhicule périodiquement un trafic vers la racine du réseau, qui doit pouvoir être acheminé en moins de 2 secondes. Nous avons donc procédé à la configuration de l'implémentation de MaCARI pour répondre aux exigences de l'application. La mise en œuvre de la solution a été effectuée dans une ancienne mine d'or canadienne dans le cadre du projet DAPOI mené par le LRTCS de Val-d'Or. Nous avons d'abord réalisé une étude de portée, puis nous avons choisi de faire deux déploiements des cartes (selon une topologie linéaire) couvrant respectivement 700 et 640 mètres de galeries. L'évaluation de la maquette de la solution dans la mine souterraine CANMET montre que l'application implémentée sur la couche MaCARI est fiable et peut supporter la présence d'un trafic additionnel de type non prioritaire. MaCARI permet de fournir à l'application la latence exigée (inférieur à $1,476$ secondes) et une faible consommation énergétique moyenne ($5,02$ mW pour une feuille et $23,04$ mW pour un coordinateur).

6.2 Perspectives

La poursuite de ce travail pourrait consister à implémenter l'ensemble des fonctionnalités de MaCARI. L'implémentation de MaCARI sur cartes TelosB dispose de quelques fonctionnalités qui demeurent statiques (comme la gestion de la topologie et des mécanismes d'association et de désassociation des nœuds) et d'autres qui sont actuellement non implémentées (comme la gestion des GTS). D'autre part, l'implémentation actuelle ne supporte que des valeurs de R_m inférieure ou égale à 2 (pas plus de deux routeurs fils) : pour chaque fils coordinateur, l'espace mémoire nécessaire à la gestion d'une file d'attente (pouvant contenir n trames de 128 octets) doit être réservé. Si l'on

souhaite que l'implémentation supporte une topologie avec un paramètre R_m strictement plus grand que 2, il faudra considérer les limites de l'espace mémoire disponible sur une carte TelosB. Une solution pourrait être d'implémenter une gestion dynamique de la capacité des files d'attente. Par la suite, il serait envisageable d'implémenter les couches supérieures du projet OCARI afin de bénéficier des performances de l'ensemble de ses principes.

Il serait également intéressant de proposer un modèle analytique permettant de borner la quantité de trafic (prioritaire et non-prioritaire) pouvant être absorbé pendant un cycle selon la topologie (c'est à dire le nombre de trames générées et la position dans l'arbre des sources de trafic, ceci en fonction du dimensionnement des différentes périodes). Une étude approfondie du trafic lors de chacune des périodes d'activité pour des topologies significatives permettrait de mieux situer les capacités du réseau. Il serait alors plus facile d'identifier la configuration optimale à adopter pour répondre aux besoins applicatifs.

D'autre part, une étude sur la robustesse de la solution dans la mine permettrait d'identifier les phénomènes pouvant perturber son fonctionnement (comme l'épuisement des batteries ou les perturbations liées au passage des véhicules et à l'activité de forage dans la mine).

D'un point de vue applicatif, il serait intéressant de réaliser une interface générique entre les réseaux de capteurs sans fil et le centre de contrôle de la mine. Cette étape permettrait à l'application de contrôle de la ventilation (ou à toute autre application basée sur les réseaux de capteurs sans fil) d'être pilotée de manière centralisée. De cette manière, nous pourrions proposer une solution modulaire permettant d'accueillir plusieurs applications industrielles. Le logiciel pourrait disposer d'une interface graphique permettant à un contrôleur humain de visualiser l'état de la mine en temps réel.

Annexe A

Correspondance technique

Cette annexe présente une partie de ma correspondance (entre décembre 09 à juin 2010) avec Jan Hauer, l'auteur principal de TKN 15.4, . Seul le contenu technique des échanges est retranscrit.

>Lionel Pascal à tinyos-help

I'm trying to figure out how to implement an experimental MAC protocol on TelosB. I came to several conclusions that might be wrong. Could you please help me to check that ?

In view of the TKN15.4 MAC (an implementation of IEEE 802.15.4 for TelosB) and the native B-MAC, it seems that it's likely impossible to implement a MAC layer without a PHY layer adaptation. It's probably I think due to the MAC functionalities involved in the CC2420 chip. Am I wrong ?

>Jan Hauer à Lionel Pascal, tinyos-help

If your MAC cannot be built on top of the existing CC2420 components (either tos/chips/cc2420 or tos/chips/c2420_tkn154) then I guess you have to dig into the CC2420 code and adapt it for your MAC. There is no chip-independent radio abstraction below

AM, every radio has its own specific MAC implementation and usually the radio driver is implemented so that it supports (at least) this MAC.

>Lionel Pascal à tinyos-help

Actually the MAC Iayer I'm working on has some similarities with 802.15.4. However, I think the TKN15.4 implementation might probably not be used in my case. What do you think about the Mac Architecture Layer proposed by Stanford university (published in Sensys2007) ?

I'm not opposed to prototyped code but i haven't found any MAC layer implemented (except the natives one) with this tool. Do you think 802.15.4 could have been realized using Stanford's MAL

>Jan Hauer à Lionel Pascal, tinyos-help

You mean this paper : http ://www.cse.wustl.edu/ lu/papers/sensys07.pdf ? If the MLME primitives can be replaced by their control interface, the nonbeacon-enabled mode could work. About the beacon-enabled mode I'm not sure, it depends on the semantics of their AsyncSend /AsyncReceive interfaces (e.g. if the time between call AsyncSend.send() and the first symbol being output by the radio is deterministic).

>Lionel Pascal à tinyos-help

I have got another question concerning the implementation of 802.15.4 in the Hurray University in Porto.

(http ://www.open-zb.net/publications/HURRAY_TR_061106_An_IEEE_802.15.4_protocol_implementation%20_in_nesCTinyOS_%20Reference_Guide_v1.2.pdf)

Could you tell me the main difference between the TKN and the Hurray 802.15.4 version ? From my point of view, I think that TKN code aims to reach a certain performance level while Hurray's code focuses on standard's functionnalities ... am I wrong ?

>Jan Hauer à Lionel Pascal, tinyos-help

They are both 802.15.4 MAC implementations. We agreed to merge them into one single version a while ago, but have not finished the process, so you can still use them separately. There are some differences in terms of implemented functionality (e.g. TKN15.4 doesn't support GTS, open-zb MAC doesn't support nonbeacon-enabled mode).

>Lionel Pascal à tinyos-help

Finally, I choose to base my implementation on TKN15.4. First I would like to trully commend your work : TKN15.4 is implemented with serious and exactness. I wish I would manage to implement my MAC protocol with the same token mecanism but i'm facing a problem : the "ResourceTransfer" interface, to the best of my knowledge, can be used only once in a module.

I would like that a module could distribute token as appropriate to 1 to n modules. I saw that, in a previous version of TKN15.4, transferto had a parameter to set the target module (like dispatchSlottedCSMA module) but in latest version, transferto as been modified and multiple distribution appears to be avoid...

I would be very thankfull if you could tell me your point on this question and eventually some indications for modification of the Resourcetransfer module ?

>Jan Hauer à Lionel Pascal, tinyos-help

Hi Lionel, it is like you say : in a previous version it was possible to transfer the token to more than one other component. I think this was done via parameterized interfaces, where the interface parameter defined the destination component (but I'm not sure anymore). This was error prone, because I had to make sure that the mapping between interface ID and wiring was consistent. Since in the current architecture there is always only one "next" component to tranfer the token to I changed this to the simpler version.

If you want to change it, I'd recommend you take a look at the old code (like you have done), but I'd recommend you not to use parameterized interfaces, but instead use single interfaces, i.e. in the implementation you transfer a token with TokenTransferToComponentX.tranfer(), instead of TokenTransferToComponent.transfer[ID_X]().

In the configuration you need a separate wiring for every interface that the component can pass the token to, but you can make less errors, because the interface already includes the component name where you pass the token too. I hope this helps.

>Lionel Pascal à tinyos-help

I made some changes but, at this point, I have put the target ressource ID in argument of the TransferClientP.transfer function. It works for now, later, i will apply your recommandation.

I have a question concerning the evaluation of this architecture. Actually, Perspectives in your technical report mention that performances evaluations should be the next step of this work. do you know if any results concerning the reactivity of this implementation on telosb have been published ? I ask this because I've noticed that some events were reported with some delay : for example the beacon notify event in testSyncStart comes with a significiant delay.

>Jan Hauer à Lionel Pascal, tinyos-help

We have not yet done a through performance evaluation of the stack, i.e. nothing has been published yet. About the delay of the beacon notify : before the event is signalled to the app, there are some tasks posted + executed, so there are no real time guarantees here (and the standard doesn't require it). E.g. the time the next higher layer gets the event depends on things link CPU speed, or number of tasks in the task queue, etc. which are things out of scope of the implementation.

>Lionel Pascal à tinyos-help

I see that there was a DebugP module in the last version that is no longer available (only the dummie appears in my tkn version). could you give me some "best practice" to implement a "debug to standard output" which involve no lag on the mote performance or behaviour ?

>Jan Hauer à Lionel Pascal, tinyos-help

There should be a tinyos-2.x/tos/lib/mac/tkn154/DebugP.nc. It will be activated (wired) only if you pass the debug option to the make system, otherwise the dummy component will be used. I guess any debugging technique will create some performance loss. There should be slightly less overhead if you use dbg_serial() than printf (but dbg_serial() is also less flexible). Take a look how it is implemented to understand what the overhead is. Since the data is written over UART/USB only when you call printfflush() or dbg_serial_flush(), this is how you can influence when the most overhead will happen.

>Lionel Pascal à tinyos-help

I 'm currently about to implement routing on my experimental MAC layer and i'm facing troubles with multi hop mechanisms. In your implementation of 802.15.4, in DataP module, you use a pool to store txframe_t type messages. The txframe stored is composed with pointers on the frame buffer. So, if 2 messages are generated during the inactive period, 2 txframes will be loaded into the pool. However , as there is only one frame buffer, when two successive data request are launched with no immediate transmission, does that makes sense to think that the first message payload will be erased by the second one ?

>Jan Hauer à Lionel Pascal, tinyos-help

Pool is used to dynamically allocate some memory. When a packet is given to DataP it will be encapsulated in a structure ieee154_txframe_t. For every packet there will be a separate allocation, when the packet has been sent ieee154_txframe_t gets deallocated. Similar to malloc and free.

>Lionel Pascal à tinyos-help

I'm facing trouble with the loading phase done by the RadioTX interface before a transmission. Actually, loading a frame is taking twice the time I could have expected : 2ms for 29 octets, around 8ms for a 127 frame (see attachment).Could you please give me hints to understand the possible reason of these values ? (resources, bus latency between

the microchip and the CC2420 module)

I see there are some comments in the CC2420TransmitP module :

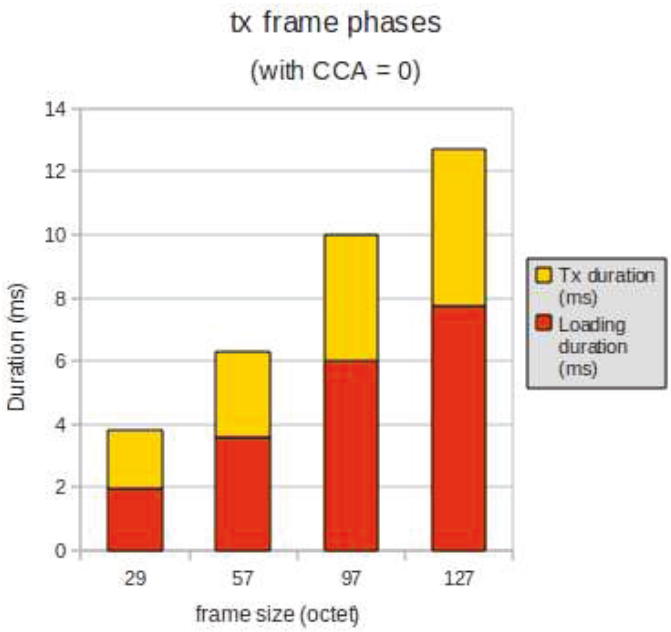
```
/** The TXFIFO is used to load packets into the transmit
 buffer on the chip */
 async event void TXFIFO.writeDone( uint8_t* tx_buf, uint8_t tx_len,
 error_t error ) {
  call CSN.set();
   if (tx_buf == &m_txFrameLen){
    // until here: 1.65 ms
    call CSN.clr();
    call TXFIFO.write( m_data->header->mhr, m_data->headerLen );
    return;
   } else if (tx_buf == m_data->header->mhr) {
    // until here: 2.2 ms
    call CSN.clr();
    call TXFIFO.write( m_data->payload, m_data->payloadLen );
    return;
   }
    // until here: 4.6 ms (no DMA on USART0)
    // until here: 3.3 ms (with DMA on USART0)
```

Do you have an idea of the context of these measurement ?

Do you think if there is a way to improve loading time ?

>Jan Hauer to Lionel Pascal

The majority of the delay is due to sending the data over the SPI bus which connects the MCU with the radio. On telosb the default clock speed of the bus is 1 MHz, i.e. 1 microsecond per bit. However, there is some extra overhead, because there is an interrupt on the MCU for every few bytes that have been transmitted. There are several things to make it faster :



– if you don't want to use DMA you could increase "SPI_ATOMIC_SIZE" in tinyos-2.x/tos/chips/msp430/usart/Msp430SpiNoDmaP(less interrupts)
– enable the DMA for the SPI bus (see ln. 61 in tinyos-2.x/tos/chips/msp430/usart/Msp430Spi0C)
– increase the clock speed of the SPI bus to 2 MHz (only works on telosb with ROSC, e.g. tmote sky). In MspDcoSpec.h use this :

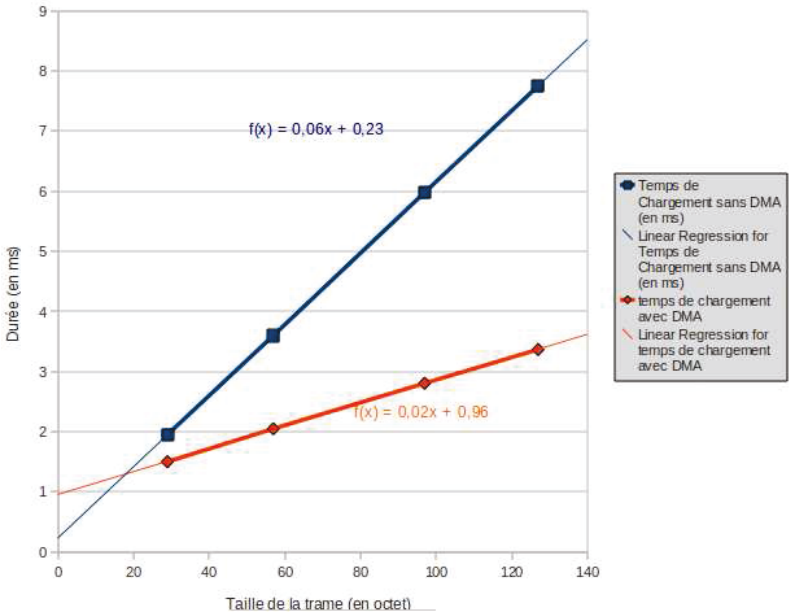
```
#define TARGET_DCO_KHZ 8192 // the target DCO clock rate in binary
kHz and make sure Dco is properly calibrated:
(tinyos-2.x/tos/chips/msp430/timer/Msp430ClockP)
event void Msp430ClockInit.setupDcoCalibrate()
 {
   call Msp430ClockInit.defaultSetupDcoCalibrate();
#if defined(MSP430_MCLK_8MHZ)
   BCSCTL2 = 0x01; // external resistor
#endif
 }
 event void Msp430ClockInit.initClocks()
 {
 // BCSCTL1
```

```
   // .XT2OFF = 1; disable the external oscillator for SCLK and MCLK
   // .XTS = 0; set low frequency mode for LXFT1
   // .DIVA = 0; set the divisor on ACLK to 1
   // .RSEL, do not modify
   BCSCTL1 = XT2OFF | (BCSCTL1 & (RSEL2|RSEL1|RSEL0));
   // BCSCTL2
   // .SELM = 0; select DCOCLK as source for MCLK
     // .DIVM = 0; set the divisor of MCLK to 1
   // .SELS = 0; select DCOCLK as source for SMCLK
       // .DIVS = 0; set the divisor of SMCLK to 2
   // .DCOR = 0; select internal resistor for DCO
#ifdef MSP430_MCLK_8MHZ
   BCSCTL2 = 0x01; // SMCLK = DCO = 8MHz, external resistor
#else
  BCSCTL2 = 0x00; // SMCLK = DCO
#endif
    // IE1.OFIE = 0; no interrupt for oscillator fault
   CLR_FLAG( IE1, OFIE );
  }
```

About my comments in the code : I can't remember what configuration I used. Please let me know how much you can decrease the delay.

>Lionel Pascal to Jan Hauer

1. Enabling DMA : I've done some evaluation (see attachment, in blue with DMA disable, in orange with DMA enable). My interpretation is that with DMA, sending DATA from the MCU to the radio is 3x more efficient. However it implies an additional time constant of 0,96 ms (against 0.25ms without DMA) whatever the size of the frame. I wish you could help me to understand this 1ms delay and tell me if my interpretation is correct.
2. increase SPI_ATOMIC_SIZE : I have read a discussion between Eric Decker and you on this link http ://mail.millennium.berkeley.edu/pipermail/tinyos-devel/2008-



June/002989.html and this parameter seems to possibly generate bugs and imprecisions...

3. increase the clock speed of the SPI bus to 2Mhz : I've compared the code indications of your previous message with the official tkn version. Should I have to copy it to the right place and set "#define MSP430_MCLK_8MHZ" ? according to the naming of this flag, don't we increase the DCO to 8Mhz ?

>Jan Hauer to Lionel Pascal

1. There is some constant delay, because the DMA needs to be configured for the operation (sending the bytes over SPI). Take a look at the DMA code and MSP430 user guide to understand what is happening.
2. The problem is that it runs in atomic and may block the system for some time. Depends if you have other critical tasks running.
3. you can use it to replace the default code, MSP430_MCLK_8MHZ should be defined.

>Lionel Pascal to Jan Hauer

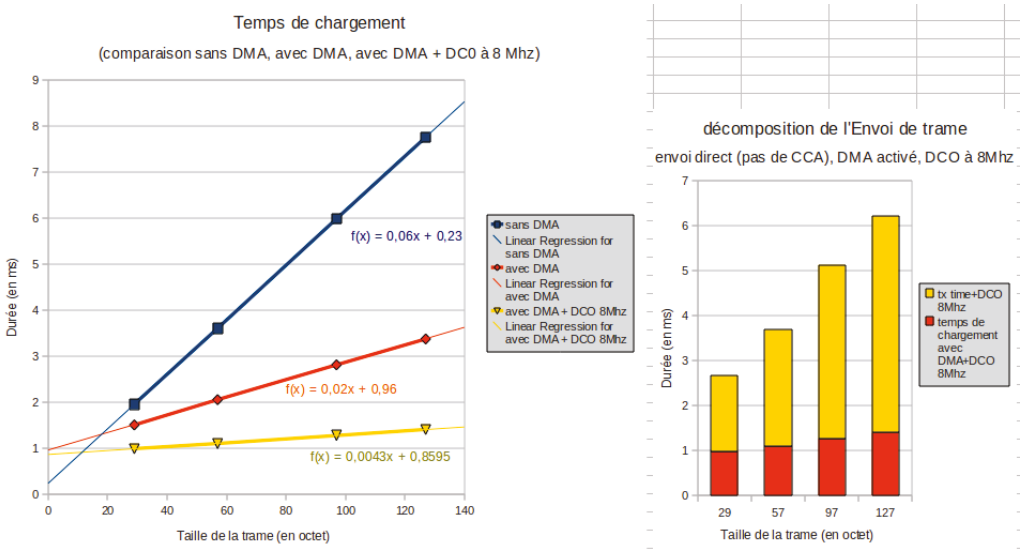
I've try these settings with DMA enabled :

```
#define TARGET_DCO_KHZ 8192 (in MspDcoSpec.h)
```

`and BCSCTL2 = 0x00 (in Msp430ClockP)`

and the performances are better (see attachment)). what TARGET_DCO_KHZ flags implies ? (I mean why TARGET ?)

As you can see on the graphes, these combined settings (DMA + DCO at 8Mhz) greatly improves loading time. don't you think these settings should be set in the telosb platform ?



>Jan Hauer to Lionel Pascal

The target DCO frequency actually involves some calibration process in software, therefore target is sort of the "goal" of this calibration process.

>Lionel Pascal to Jan Hauer

When a device is trying to send something (state =WAIT_FOR_TXDONE) in the CAP period, I see that it can't receive any msg during the backoff duration.

I have tried a simple test : a deviceA sends succesively 2 messages (msg1 and msg2) to a deviceB, at the reception of the first message deviceB sends (relay) msg1 :

it appears that the receive event of msg is not signaled by the deviceB while it a is well received on the sniffer... i think that radio is on TX mode all this time so that's why deviceB don't receive msg2... am I wrong ?

>Jan Hauer to Lionel Pascal

In general, before a transmission the radio is switched to Rx mode iff a CCA is required. When the CCA fails (channel busy), the radio is switched off during the backoff period. If CCA succeeds then packet is sent without switching radio off in between.

Annexe B

Liste des abreviations

ANR	Agence Nationale de Recherche
AODV	Ad hoc On-Demand Distance Vector
BE	Backoff Exponent
BLE	Battery Life Extension
B-MAC	Berkeley-MAC
CANMET	Canada Centre for Mineral and Energy Technology
CAP	Contention Access Period
CCA	Clear Channel Assessment
CFP	Contention Free Period
CISTER	Research Centre in Real-Time Computing Systems
CNAM	Conservatoire National des Arts et des Métiers
CNRS	Centre National de la Recherche Scientifique
CPAN	Coordinateur du PAN
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CW	Contention Window
DAPOI	Démonstration Appliquée et Prototypage d'Outils Industriels
DCNS	Direction de Constructions Navales Sextant

DCO	Digital Controlled Oscillator
DMA	Direct Memory Access
EDF	Electricité De France
EDS	Exposant de Durée d'un Slot
E-MAC	Event-MAC
EOLSR	Energy-aware OLSR
FFD	Full Function Device
GTS	Guaranteed Time Slot
IEC	Commission Electrotechnique Internationale
IEEE	Institute of Electrical and Electronics Engineers
IFMA	Institut Français de Mécanique Avancée
INRIA	Institut National de Recherche en Informatique et Automatique
IPP-HURRAY	Institut Polytechnique de Porto - HUgging Real-time and Reliable Architectures for computing sYstems
ISA	International Society of Automation
LATTIS	LAboratoire Toulousain de Technologie et d'Ingénierie des Systèmes
LIMOS	Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes
LLC	Logical Link Control
LOS	Line Of Sight
LPL	Low Power Listening
LP-WAN	Low Power - Wireless Personal Area Network
LQI	Link Quality Indication
LRI	Laboratoire de Recherche en Informatique
LRTCS	Laboratoire de recherche Télébec en communications souterraines
MAC	Medium Access Layer
MaCARI	Mac pour oCARI
MLDE	MAC Layer Data Entity
MLME	MAC Layer Management Entity
NB	Number of Backoffs
NIB	Network layer Information Base

NLDE	Network Layer Data Entity-Service Access Point
NLME	Network Layer Management Entity
NLOS	Not Line Of Sight
NM	Network Manager
NS2	Network Simulator 2
OCARI	Optimisation des Communications Ad-hoc pour les Réseaux Industriels
OLSR	Optimized Link State Routing
PAN	Personal Area Network
PD	PHY layer Data
PHY	Physical layer
PIB	Pan Information Based
PLME	Physical Layer Management Entity
RFD	Reduced Functionality Device
RSSI	Received Signal Strength Indicator
SAP	Service Access Point
SO	Superframe Order
SPI	Serial Peripheral Interface
ST2I	Sciences et Technologies de l'Information et de l'Ingénierie
TDMA	Time Division Multiple Access
TKN	Département de recherche de l'Université Technique de Berlin
UMR	Unité Mixte de Recherche
UQAT	Université Québécoise d'Abitibi-Témiscamingue
WFD	Wireless Field Device
WPAN	Wireless Personnal Area Network
Z-MAC	Zebre-MAC

Annexe C

Glossaire

Bus de données

Liaison physique reliant plusieurs composants matériels par lequel sont echangées les informations. Un bus est caractérisé par une fréquence et le nombre de bits d'informations qu'il peut transmettre simultanément.

Interruption

Arrêt temporaire de l'exécution normale d'un processus par le microprocesseur afin d'exécuter une autre tâche.

Processus ou tâche

Struture englobant un ensemble d'instructions à exécuter.

Trame

Paquet d'informations délimitées par des fanions.

Beacon

Trame balise périodique utilisée pour délimiter un cycle et/ou définir une réference de temps dans un réseau. Un beacon peut éventuellement contenir des informations fonctionnelles.

Slot

Intervalle de temps utilisé comme unité.

Médium

Support par lequel les informations sont transmises

Topologie

Architecture et organisation d'un réseau.

Qualité de service

Capacité à véhiculer dans de bonnes conditions un type de trafic donné, en termes de disponibilité, débit, délais de transmission, taux de pertes de trames...

Routage

Mécanisme par lequel des chemins sont sélectionnés dans un réseau pour acheminer les données d'un expéditeur jusqu'à un ou plusieurs destinataires.

Open-Source

Terme désignant les programmes et logiciels dont la licence respecte des critères établis par l'Open Source Initiative : possibilité de libre redistribution, d'accès au code source et de travaux dérivés.

Latence

Temps nécessaire à un paquet de données pour passer de la source à la destination à travers un réseau.

Références

[1] J. Hart and K. Martinez. Environmental sensor networks : A revolution in the earth system science ? *Earth-Science Reviews*, pages 177–191, 2006.

[2] The ISA100 standards website, 2008. http ://www.isa.org/isa100.

[3] Le projet OCARI. http ://ocari.lri.fr.

[4] T. Dang, C. Devic, and et al. Ocari : Optimization of communication for ad hoc reliable industrial networks. *IEEE INDIN*, january 2008.

[5] ANR. Agence Nationale de Recherche. http ://www.agence-nationale-recherche.fr.

[6] Network Simulator 2, 2002. http ://www.isi.edu.nsnam/ns.

[7] Telit. http ://www.telit.com/.

[8] G. Chalhoub. MACARI : Une méthode d'accès déterministe et économe en énergie pour les reseaux de capteurs sans fil, 2009. thèse au LIMOS.

[9] IEEE 802.15. Part 15.4 : Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs). Standard 802.15.4 R2006, ANSI/IEEE, 2006.

[10] A. Burr. *Modulation and Coding for Wireless Communications*. Pearson Education, 2001.

[11] A. Koubaa, M. Alves, and E. Tovar. A comprehensive simulation study of slotted CSMA/CA for IEEE 802.15.4 wireless sensor networks. *WFCS*, pages 1–10, july 2006.

[12] ZigBee. ZigBee Specification. Standard Zigbee 053474r17, ZigBee Standards Organization, January 2008.

[13] ZigBee Alliance. The ZigBee Alliance website. http ://www.zigbee.org.

[14] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-demand Distance Vector (AODV) routing. Request For Comments 3561, IETF, July 2003.

[15] WirelessHART, HART protocol revision v.7.0, 2007. http ://www.hartcomm.org.

[16] T. Nieberg, S. Dulman, P. Havinga, L. Van Hoese, and J. Wu. Collaborative algorithms for communication wireless sensor networks. In *Ambient Intelligence : Impact on embedded Systems*. Kluver academic publishers, November 2003. ISBN 1-4020-7668-1.

[17] J. Polastre, J. Hill, and D. Culler. Versatile low-power medium access for wireless sensor networks. In *ACM Sensys*, 2004.

[18] I. Rhee, A. Warrier, M. Aia, and M. J. Z-MAC : a hybrid mac for wireless sensor networks. In *ACM Sensys*, November 2005.

[19] H. Zayani, R. Ben Ayed, K. Djouani, and K. Barkaoui. Eco-mac : an energy-efficient and low-latency hybrid mac protocol for wireless sensor networks. In *PM2HW2N*, 2007.

[20] G. Chalhoub, E. Livolant, A. Guitton, A. van den Bossche, M. Misson, and T. Val. Specifications and evaluation of a MAC protocol for a LP-WPAN. *Ad Hoc & Sensor Wireless Networks journal*, 2009.

[21] G. Chalhoub, A. Guitton, F. Jacquet, A. Freitas, and M. Misson. Medium access control for a tree-based wireless sensor network : Synchronization management. In *Wireless Days (IFIP Wireless Days)*, November 2008.

[22] G. Chalhoub, A. Guitton, and M. Misson. MAC specifications for a WPAN allowing both energy saving and guaranted delay - Part A : MaCARI : a synchronized tree-based MAC protocol. In *IFIP WSAN*, 2008.

[23] S. Mahfoudh and P. Minet. EOLSR : an energy efficient routing protocol in wireless ad hoc and sensor networks. *Journal of Interconnection Networks*, 9(4), 2008.

[24] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized Link State Routing protocol. RFC 3626, IETF, 2003.

[25] Crossbow TelosB. TelosB datasheet. www.xbow.com.

[26] P. Levis. TinyOS Programming. Technical report, june 2006. guide technique.

[27] D. Gay and P. Levis. nesC 1.1 Language Reference Manual, may 2003.

[28] Chipcon. Chipcon CC2420. http ://focus.ti.com/docs/prod/folders/print/cc2420/html.

[29] R. Polastre, J. Szewczyk and D. Culler. Telos : enabling ultra-low power wireless research. In *Information Processing in Sensor Networks*, 2005.

[30] The TinyOS website. http ://www.tinyos.net.

[31] A. Cunha, M. Alves, and A. Koubâa. An IEEE 802.15.4 protocol implementation (in nesC/TinyOS) : Reference guide v1.2. Technical report, Polytechnic Institute of Porto, May 2007.

[32] R. Severino. On the use of IEEE 802.15.4/ZigBee for time-sensitive wireless sensor network applications, 2008.

[33] A. Cunha. On the use of IEEE 802.15.4/ZigBee as federating communication protocols for wireless sensor networks, 2007.

[34] J-H. Hauer. TKN15.4 : An IEEE 802.15.4 MAC Implementation for TinyOS 2. Technical report, TKN. Technical University Berlin, march 2009.

[35] Texas Instrument. MSP430 datasheet, june 2004.

[36] Direct Memory Access. http ://en.wikipedia.org/wiki/Direct_memory_access.

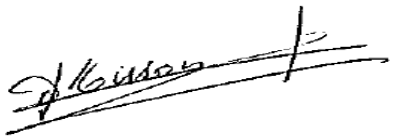
[37] S. Roicomte. Etude des réseaux de capteurs sans fil, 2010.

[38] IEEE 802.11. Part 11 : Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. Standard 802.11 R2003, ANSI/IEEE, 1999.

[39] Site web LRTCS. http ://lrtcs.ca.

[40] CANMET. http ://canmetenergy-canmetenergie.nrcan-rncan.gc.ca/fra.

Vu, le Professeur Principal



M. MISSON

Le Directeur du Cnam d'Auvergne



G. FLEURY

A Aubière, le 17 mai 2011

Résumé

Les réseaux de capteurs sans fil constituent une réponse nouvelle adaptée aux besoins de certaines applications industrielles : selon le profil de l'application, ces réseaux doivent fournir les fonctionnalités nécessaires et s'adapter à des exigences de latence, de qualité de service et/ou d'autonomie principalement. Le projet OCARI propose une pile protocolaire pour les réseaux de capteurs sans fil utilisant la méthode d'accès MaCARI, qui permet de répondre à certains besoins applicatifs industriels, notamment en termes d'économie d'énergie et de déterminisme. MaCARI est implémentée sur une plateforme matérielle propriétaire Telit impliquant des limitations quant à la propriété intellectuelle du code.

Ce mémoire décrit d'abord la réalisation de l'adaptation et du portage de MaCARI sur la plateforme open-source TelosB. Puis, il détaille le développement et la mise en œuvre d'une application de surveillance basée sur cette implémentation destinée à l'industrie minière et testée dans une mine-laboratoire canadienne.

Mots clés : Réseaux de capteurs sans fil, méthode d'accès au médium, qualité de service, mine souterraine.

Abstract

Wireless sensor networks are a novel adapted response for most industrial applications : according to the profile application, these networks must provide the needed functionalities and satisfy the overall requirements such as low latency, good quality of service and autonomy. The OCARI project proposes a new stack architecture using MaCARI as a medium access method. MaCARI is a deterministic protocol that proves the network performance in terms of energy savings. MaCARI is implemented on Telit motes involving limitations on the source code property.

This thesis studies first the adaptation of MaCARI on TelosB motes, an open-source platform. Then, based on this study, the implementation of monitoring application are examinated for the mining industry and evaluated in a canadian mine.

keywords : Wireless sensor networks, medium access method, quality of service, underground mine.