

THÈSE

Présentée à

L'UFR DES SCIENCES ET TECHNIQUES
DE L'UNIVERSITÉ DE FRANCHE-COMTÉ

Pour obtenir le

**GRADE DE DOCTEUR
DE L'UNIVERSITÉ DE FRANCHE-COMTÉ**

Spécialité Informatique

Conception d'une architecture hiérarchique de réseau de capteurs pour le stockage et la compression de données

Par

**Medetonhan Shambhalla Eugène William
PAMBA CAPO-CHICHI**

Soutenue le 11 Mars 2010 devant le jury :

Rapporteurs :	CONGDUC PHAM	Professeur, Université de Pau et des Pays de l'Adour
	BERNARD TOURANCHEAU	Professeur, Ecole Normale Supérieure de Lyon
Directeurs :	HERVÉ GUYENNET	Professeur, Université de Franche-Comté
	JEAN-MICHEL FRIEDT	Ingénieur de recherche, Sensor
Examineurs :	MICHEL DE LABACHELERIE	Directeur de recherche CNRS, Université de Franche-Comté
	MICHAËL KRAJECKI	Professeur, Université de Reims Champagne-Ardenne
	JEAN-CHRISTOPHE LAPAYRE	Professeur, Université de Franche-Comté
	PASCAL LORENZ	Professeur, Université de Haute Alsace

*Je dédie cette thèse principalement à mes Parents Laurence et Edouard pour leur Amour profond
qui a toujours su m'éclairer et le goût du travail bien accompli qu'ils ont su m'inculquer.
A mes deux Sœurs Sophie et Grâce que j'aime tant pour leurs encouragements, leurs conseils et
leurs Prières.*

*A la mémoire de ma Tante et Sœur Nadège, Nana. Que ton sourire et ton éternelle bonne humeur
continuent à briller dans nos Coeurs et illuminer le Ciel! Je tiendrai la promesse que je t'ai faite.*

*A ses Enfants Leo Neil et Yorick-Xavier,
A mes neveux Noa-Edouard et Thalya,
A ma Tante Léonie pour ses encouragements réguliers.*

*C'est en pensant à vos tous que j'ai puisé la Force nécessaire pour conduire ce travail.
Je vous aime...*

Remerciements

Ce travail a été réalisé au laboratoire d'informatique de Franche-Comté (LIFC), sous la bienveillante direction de Monsieur le Professeur Hervé Guyennet, auquel j'adresse mes très sincères remerciements pour sa disponibilité, son encadrement et les moyens matériels qu'il a mis à ma disposition.

J'exprime ensuite ma profonde gratitude à Monsieur Jean-Michel Friedt, mon co-directeur de thèse du département Temps-Fréquence du laboratoire FEMTO/ST, pour les nombreux conseils qu'il m'a donnés, pour le temps considérable qu'il m'a consacré et pour la confiance totale qu'il m'a témoignée.

Que Monsieur le Professeur Congduc Pham et Monsieur le Professeur Bernard Tourancheau me permettent de leur redire ma reconnaissance pour le temps consacré à mes rapports de thèse malgré leur emploi de temps surchargé.

Je remercie également le Directeur de recherche CNRS, Monsieur Michel De Labachellerie, Monsieur le Professeur Michaël Krajecki et Monsieur le Professeur Pascal Lorenz d'avoir participé à ce jury. Je leur sais gré de l'intérêt porté à cette thèse en acceptant de l'examiner.

Je tiens à remercier vivement Monsieur le Professeur Jean-Christophe LAPAYRE qui m'a fait l'honneur de présider ce jury.

Ma dette de reconnaissance est très lourde envers Monsieur Craig Duffy, *Principal Lecturer* à *The Faculty of Computing, Engineering and Mathematical Sciences* de l'université de West England à Bristol en Angleterre, pour son invitation de trois mois; à Monsieur Ian Johnson, *Lecturer* dans la même université, pour son encadrement pointu pendant ma visite de recherche à Bristol et pour la publication effectuée ensemble; à tous les membres du *Bristol Institut of Technology* pour l'accueil chaleureux qu'ils m'ont réservé.

Je remercie ensuite mon collègue et ami Monsieur Laurent Steck pour ses conseils, pour nos longues discussions et pour les moyens matériels mis à disposition; à Monsieur Jean-Michel Caricand pour sa disponibilité et ses blagues légendaires, à Christine Bigey pour son attention et sa gentillesse et à Brigitte Bataillard pour son aide et sa courtoisie.

Je ne saurais oublier tout le département GLT de l'IUT de Vesoul; Lucia, Pierre, Anne-Chantal, Thomas, Marie-Aimée, Marlène et tous les autres pour leur bonne humeur quotidienne et leur accueil chaleureux.

Je remercie également tous ceux qui ont su créer autour de moi une ambiance fraternelle; Liliane, René-Pierre, Andjouz, Bello et Felix,

A mon cousin et ami Raoul pour ses encouragements et sa présence le jour de ma thèse,

A Christine, Nènè, pour ses pensées positives et ses encouragements,

A mes collègues; David Martins, Le Hung-Cuong, Kamal Beydoun et Mohamed Leshaini pour nos nombreuses discussions sur les réseaux de capteurs; A mes étudiants Gwenael Goavec-Merou et Geoffroy Weisenhorn pour le formidable travail effectué ensemble,

A mon cousin Stéphane pour sa disponibilité,

A Linda pour tout.

Table des matières

Introduction générale

9

Partie I Etat de l'art

Chapitre 1

Généralités sur les réseaux de capteurs
--

1.1	Le capteur	17
1.1.1	Architecture d'un capteur	17
1.1.2	TinyOS : environnement de développement dédié aux capteurs	18
1.2	Etat de l'art sur les réseaux personnels sans fil	21
1.2.1	IRDA	21
1.2.2	IEEE 802.15.1/Bluetooth	21
1.2.3	Wibree - Ultra Low Power (ULP) Bluetooth	23
1.2.4	HOME RF	24
1.2.5	IEEE 802.15.3/UWB	24
1.2.6	IEEE 802.15.4/ZigBee	24
1.3	Les réseaux <i>ad hoc</i>	27
1.3.1	Les protocoles hiérarchiques	28
1.3.2	Les protocoles géographiques	29
1.3.3	Les protocoles à plat	29
1.4	Les réseaux de capteurs sans fil	29
1.4.1	Le déploiement	30
1.4.2	La couverture de surface et la collecte de données	30
1.4.3	La contrainte énergétique	31
1.5	Les applications utilisant les réseaux de capteurs	32
1.5.1	Les applications militaires	32
1.5.2	Les applications médicales	32
1.5.3	Les applications liées à la sécurité des infrastructures	32

1.5.4	La domotique	33
1.5.5	Les applications environnementales	33
1.6	Conclusion	33

Chapitre 2

Les plateformes et les projets de réseaux de capteurs

2.1	Les plateformes de réseaux de capteurs	36
2.2	Les projets	41
2.2.1	The Wired Integrated Network Sensors (WINS) 1993-1998	41
2.2.2	Smart Dust (Berkeley) 1995-2001	42
2.2.3	Smart-Its	43
2.2.4	The ZebraNet Wildlife Tracker	44
2.2.5	TurtleNet	46
2.2.6	Volcano	47
2.2.7	A macroscope in the redwoods	48
2.2.8	Glacsweb	50
2.2.9	Sensorscope	51
2.2.10	Récapitulatif sur l'évolution des plateformes	52
2.2.11	Le programme HYDRO-SENSOR-FLOWS (2006-2010)	54
2.3	Synthèse sur les projets de réseaux de capteurs	56
2.4	Les architectures hybrides	56
2.4.1	SensEye	57
2.4.2	Surveillance d'un site archéologique	60
2.5	Conclusion	60

Partie II Mes contributions 63

Chapitre 3

Une architecture hiérarchique et générique de réseau de capteurs

3.1	Présentation de l'architecture générale	66
3.2	Évaluation des performances du niveau supérieur de notre architecture	67
3.2.1	La carte Fox	67
3.2.2	Le module d'évaluation énergétique	68
3.2.3	Description de l'architecture réseau pour le transfert d'images	69
3.2.4	Résultats expérimentaux	71
3.3	Synthèse sur le niveau supérieur de notre architecture	74

3.4	Présentation du moyen de communication entre les différents niveaux	75
3.4.1	La Tmote Sky	75
3.4.2	Le protocole IEEE 802.15.4	76
3.5	Notre architecture hiérarchique et générique de réseaux de capteurs	78
3.5.1	Implémentation du protocole IEEE 802.15.4 sur notre plateforme	79
3.5.2	Evaluation du compromis énergie/délai du protocole IEEE 802.15.4	83
3.5.3	Introduction de l' α -pénalité	85
3.6	Conclusion	88

Chapitre 4

Nouvelles techniques de compression de données dans les réseaux de capteurs

4.1	Les algorithmes de compression	91
4.1.1	Les algorithmes avec pertes	91
4.1.2	Les algorithmes sans pertes	92
4.2	L'algorithme S-LZW	93
4.3	L'algorithme RLE (Run-Length Encoding)	94
4.4	Première contribution : l'algorithme K-RLE (K-Run-Length Encoding)	96
4.4.1	Principe	96
4.4.2	Résultats expérimentaux	98
4.5	Synthèse sur l'algorithme K-RLE	111
4.6	Deuxième contribution : l'algorithme Aganmà	112
4.6.1	Principe	112
4.6.2	Résultats expérimentaux	114
4.7	Synthèse sur l'algorithme Aganmà	116
4.8	Conclusion	117

Chapitre 5

Stockage de données dans les réseaux de capteurs

5.1	Présentation générale	121
5.2	Communication synchrone SPI et carte mémoire	122
5.2.1	Le Bus SPI	122
5.2.2	Support de TinyOS	122
5.2.3	Fonctionnement de la SD	123
5.2.4	Implémentation	123
5.3	Stockage en mémoire non volatile	124
5.3.1	Structure de la FAT16	125
5.3.2	Problématique	126

5.3.3	Implémentation	126
5.4	Description du module GPS	132
5.4.1	Structure des trames NMEA	132
5.4.2	Fonctionnement	132
5.4.3	Module GPS	132
5.5	Evaluation du module de stockage	133
5.6	Test et performances	134
5.6.1	Objectif	134
5.6.2	Conditions de l'expérience	134
5.6.3	Résultats expérimentaux	136
5.7	Conclusion	140

Conclusion et perspectives	141
-----------------------------------	------------

Bibliographie	1
----------------------	----------

Travaux de recherche	9
-----------------------------	----------

Bibliographie	11
----------------------	-----------

Table des figures

1	Architecture hiérarchique d'un réseau de capteurs	11
1.1	Les composants d'un capteur	18
1.2	Pile Bluetooth	22
1.3	Représentation des différents modèles réseau de Bluetooth	23
1.4	La pile du protocole IEEE 802.15.4/ZigBee	25
1.5	Comparaison énergétique entre transceiver et micro-processeurs [28]	26
1.6	Architecture générale d'un réseau de capteurs sans fil [29]	26
1.7	Classifications des protocoles dans les réseaux <i>ad hoc</i>	28
1.8	Evolution des technologies informatique et électronique entre 1990 - 2003 [43] . .	31
2.1	Les plateformes de réseaux de capteurs [65]	36
2.2	Tableau récapitulatif des caractéristiques des plateformes des réseaux de capteurs	37
2.3	Classification des plateformes de réseaux de capteurs	38
2.4	Sun Spot	40
2.5	LWIM III	42
2.6	Les principaux composants d'un smart dust [77]	43
2.7	WeC	43
2.8	Les motes Rene (a) et Dot (b)	43
2.9	Description de la plateforme de Zebranet	45
2.10	Zebranet	46
2.11	Photo d'une tortue faisant partie du projet Turtlenet	47
2.12	Architecture de surveillance des volcans	48
2.13	Capteur sismique et acoustique sans fil	48
2.14	Description de l'architecture générale de Macroscope	49
2.15	Photo du glacier observé à Briksdalsbreen en Norvège	50
2.16	Architecture générale du réseau	50
2.17	Architecture des capteurs (version 2) déployés dans le glacier	51
2.18	Architecture de la station de base	51
2.19	Une station de collecte SensorScope	52
2.20	Tableau récapitulatif des projets de réseau de capteurs	53
2.21	Evolution du phénomène des réseaux de capteurs [94]	54
2.22	Capteurs déployés au Spitzberg équipés d'appareils photo numérique	55
2.23	Architecture de référence pour le multimédia dans les RdC [101]	57
2.24	Architecture hiérarchique de Senseye [103]	58
2.25	Scénario de détection, reconnaissance et suivi d'objets [103]	59
2.26	Architecture générale de surveillance du site archéologique d'Agriente	61

2.27	Architecture hybride et hiérarchique pour la surveillance du site archéologique d'Agrigente	61
3.1	Notre architecture générale de réseau de capteurs	67
3.2	La carte Fox [16]	68
3.3	Schéma de l'analyseur d'énergie	68
3.4	Notre carte Fox expérimentale	69
3.5	Notre réseau point à point	70
3.6	La carte Fox, une Webcam Logitech et un dongle Bluetooth	71
3.7	Débit du grabber	72
3.8	Débit du router en utilisant le dongle de class I	73
3.9	Débit du router en utilisant le dongle de class II	73
3.10	La Tmote Sky	75
3.11	Topologies proposées par le standard IEEE 802.15.4 [114]	76
3.12	Structure d'une <i>Supertrame</i> [115]	77
3.13	La carte Fox, la Tmote Sky et le dongle IEEE 802.15.4/ ZigBee	78
3.14	Consommation énergétique de la carte Fox une courte période d'inactivité de 0,73 s	80
3.15	La mote MicaZ équipée d'un ampermètre commercial	81
3.16	Délai de la carte Fox et de la MicaZ sans période de somnolence	82
3.17	Délai de la carte Fox et de la MicaZ avec une période de somnolence de 1,7 s	82
3.18	Evaluation de la puissance totale de la carte Fox en fonction du temps de veille	84
3.19	Evaluation de la puissance totale de la carte Fox en prenant en compte l' α -pénalité	86
3.20	Evaluation de la puissance totale de la MicaZ en prenant en compte l' α -pénalité	86
3.21	Variation des temps de veille optimums de la carte Fox en fonction de l' α -pénalité	87
3.22	Variation des temps de veille optimums de la MicaZ en fonction de l' α -pénalité	87
4.1	Les variantes de RLE [125]	94
4.2	Algorithme de compression RLE [124]	96
4.3	Algorithme de compression K-RLE	97
4.4	Représentation des variations de températures de différents endroits du globe	98
4.5	Comparaison entre S-LZW et RLE	99
4.6	Comparaison entre RLE et K-RLE	100
4.7	Comparaison entre S-LZW et K-RLE	101
4.8	Représentation du taux de données modifié par 2-RLE	102
4.9	Evaluation de consommation de la phase de compression des algorithmes	103
4.10	Evaluation de consommation de la phase de décompression des algorithmes	104
4.11	Courbe typique d'écart-type d'Allan [131]	106
4.12	Variance d'Allan appliquée à différentes archives de températures	106
4.13	Tableau récapitulatif des résultats obtenus pour chaque fonction	107
4.14	Comparaison entre S-LZW, RLE et K-RLE avec différentes valeurs du paramètre K	107
4.15	Comparaison des taux de données modifiées avec les paramètres de la variance d'Allan et de l'écartype (Std-RLE)	108
4.16	Evaluation des temps de compression et du temps de transmission des données compressées	109
4.17	Evaluation de la puissance des différents transceiver [82]	110
4.18	Evaluation de l'énergie totale de l'utilisation de l'algorithme de compression	111
4.19	Tableau récapitulatif des caractéristiques des algorithmes de compression précédents	112
4.20	Comparaison entre Aganmà et les algorithmes précédents	114

4.21	Evaluation de la consommation d'énergie pendant la compression	115
4.22	Evaluation du temps de compression et du temps de transmission avec Aganmà et Std-RLE	116
4.23	Evaluation de l'énergie totale d'Aganmà et Std-RLE	117
5.1	Description du module gérant le SPI sur le MSP430	123
5.2	Interface d'utilisation du module SD	124
5.3	Structure d'une partition FAT	125
5.4	Schéma du stockage	127
5.5	Fichier de configuration	129
5.6	Accès à la carte SD	129
5.7	Schéma électronique de notre plateforme équipée d'un emplacement pour une carte mémoire secure digital (SD) et d'un récepteur GPS	131
5.8	Exemple de relevé de trames GPS	132
5.9	Validation de la portabilité de notre environnement de stockage sur TelosB (a) et MicaZ (b)	134
5.10	Photos de la salle (vue de l'entrée)	135
5.11	Photos de la salle (vue du fond)	136
5.12	Plan de la salle vu du haut	136
5.13	Variation de la tension des capteurs sur les 40 premières heures	137
5.14	Variation de la tension des 4 premiers capteurs sur toute la durée de l'expérience	137
5.15	Variation de la température	138
5.16	Variations de la température et de la luminosité	139
5.17	Variation de la luminosité	139

Introduction générale

Que nous évoquent les réseaux de capteurs ? La notion de réseau liée à ce nouveau concept est bien connue des informaticiens car elle se rattache à un domaine familier en informatique, celui des réseaux et télécommunications. Ce domaine en perpétuelle ébullition grâce aux progrès technologiques incessants s'intéresse aujourd'hui à de nouveaux petits dispositifs appelés capteurs capables d'interagir avec leur environnement. Le terme *capteur* utilisé en informatique ne représente pas seulement le composant électronique qui relève des données sur l'environnement telles que la température, l'humidité ou encore d'autres quantités physiques, mais désigne le nœud complet du réseau auquel il est associé.

Les capteurs, nouveaux venus dans les thèmes de recherche en informatique, font partie du monde électronique mais nécessitent une couche logicielle de bas niveau faisant intervenir les informaticiens. C'est pour cette raison que le laboratoire d'informatique de Franche-Comté (LIFC) et le département Temps-Fréquence du laboratoire FEMTO/ST ont pu mettre en commun leurs compétences pour développer des recherches dans ce domaine. Les informaticiens doivent donc, dans un premier temps, s'appropriier la notion de capteurs dits intelligents et des outils d'exploitation qui y sont rattachés.

Ces nouveaux petits composants à capacités réduites et limités en énergie, sont capables non seulement de collecter des informations sur leur environnement mais aussi de les stocker ou de les transmettre. Cette dernière caractéristique va permettre de distribuer leurs puissances de calcul, voire de stockage, dans un système dit collaboratif. Dans ces systèmes distribués émergents, les nœuds collaborent afin de fournir une vue détaillée de leur environnement et rappatrient les informations vers une station collectrice appelée *puits* pour un futur traitement. Il faut noter que les acteurs de ces systèmes diffèrent selon l'application visée et le lieu de déploiement. Cependant, la caractéristique commune à toutes ces applications est l'énergie qui constitue le point critique dans ces réseaux.

Dans le cadre de cette thèse, nous allons nous intéresser aux architectures capables de fournir des informations riches pour l'utilisateur final comme le multimedia. Ce type d'application de prime abord est en contradiction avec les capacités des capteurs qui offrent des performances limitées. C'est pour cette raison que nous allons utiliser dans notre architecture d'autres nœuds dotés de puissances supérieures.

Nous allons donc définir une architecture de réseaux de capteurs à plusieurs niveaux. Tout d'abord, les nœuds classiques qui s'articulent autour de microcontrôleurs à fréquence limitée et puis des plateformes plus évoluées dotées d'outils de haut niveau plus familiers aux informaticiens. En effet, la convergence entre l'embarqué et le PC de bureau a donné naissance à de nouvelles plateformes de plus haut niveau équipées de processeurs embarqués cent fois plus ra-

pides que les nœuds classiques mais avec le même facteur en terme de consommation énergétique. Nous allons donc évaluer les performances de ces plateformes de haut niveau dans le cadre du transport d'images avec de bonnes résolutions.

C'est la mise en commun de ces différents niveaux qui forme une plateforme hiérarchique où les éléments de bas niveau collectent de simples données scalaires tandis que les dispositifs de haut niveau servent de concentrateur d'informations capables de réaliser des tâches complexes et de fournir des informations riches à l'utilisateur final.

Le trait d'union entre les différents niveaux de ce type d'architecture va être la technologie sans fil. En effet, l'évolution des techniques de télécommunication et la lourdeur des liaisons filaires a poussé les industriels au développement de plusieurs technologies sans fil. Cependant, les contraintes des capteurs par rapport aux techniques de transmission, qui constituent la principale source de consommation, font qu'elles ne sont pas toutes adaptées pour ces réseaux.

Le standard IEEE 802.15.4 se démarque par rapport aux autres technologies sans fil pour les réseaux de capteurs en proposant une supertrame capable de mettre le module de communication en veille pour économiser l'énergie. Cependant, la mise en veille rend indisponible le nœud dans le réseau, ce qui nous a poussé à évaluer le compromis énergie/délai associé à cette technologie.

Dans le même temps, tandis que les plateformes de haut niveau utilisent le standard précédent pour communiquer avec celles de niveau plus bas, elles sollicitent aussi les performances d'autres technologies sans fil de plus haut débit pour transporter des données lourdes telles que du multimedia vers le puits. Nous allons donc évaluer le transport d'images à l'aide de la technologie sans fil Bluetooth qui offre un compromis intéressant entre consommation énergétique et débit offert.

Les écarts de consommation et le rôle concentrateur du niveau haut font qu'il doit être alimenté en permanence, ce qui nous a poussé à nous intéresser à des stratégies d'économie d'énergie au niveau des feuilles du réseau dont les batteries sont irremplaçables. La principale source de consommation étant la transmission de données, nous allons nous focaliser sur une stratégie permettant d'optimiser l'exploitation du canal de communication : la compression de données. Il faut noter que ce choix d'étude fait que les applications prises en compte sont de type observation c'est-à-dire centrées sur les données et non sur les événements. Nous proposons deux algorithmes de compression de données dans le cadre de cette thèse.

Les travaux sur la compression de données dans les réseaux de capteurs sont encore très rares mais cette stratégie semble être une voie prometteuse car les résultats de nos propositions ont montré les bénéfices de l'utilisation de la compression de données.

Les technologies sans fil, parce qu'elles évoluent très vite, manquent encore de maturité et sont renouvelées régulièrement. Aussi, les conditions environnementales, le coût énergétique de la transmission et le poids des données observées par rapport au débit limité des technologies dédiées aux réseaux de capteurs, nous ont poussé dans la dernière partie de cette thèse, à ignorer le module de transmission et donc à utiliser des techniques de stockage d'informations sur des supports amovibles. Ce stockage d'informations se base sur un nouvel environnement utilisant un système de fichier FAT16 et donc capable d'être utilisé sur n'importe quel système d'exploitation moderne. Malgré le fait que ce choix nécessite une intervention humaine au moment de la récupération des données présentes sur les supports amovibles, nous présentons ses bénéfices telles que la quantité

importante d'informations susceptibles d'être stockées et sa généricité.
 La Fig. 1 illustre notre architecture générale de travail.

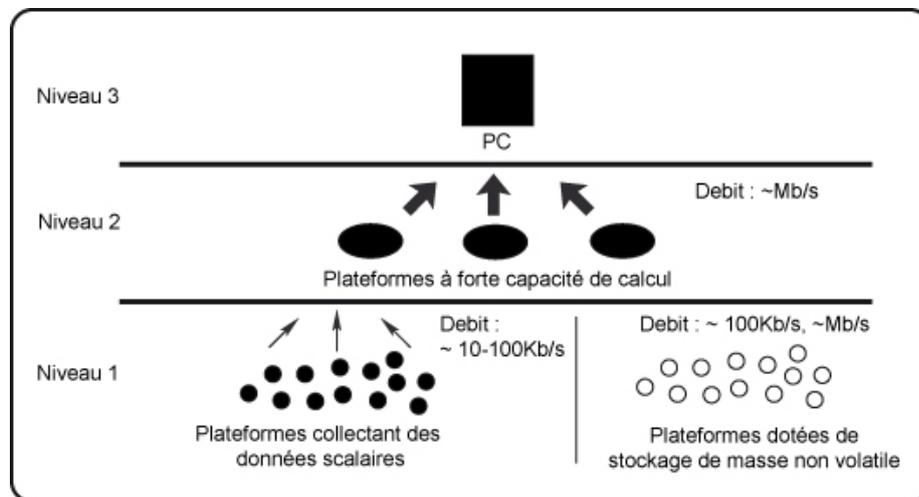


FIG. 1 – Architecture hiérarchique d'un réseau de capteurs

L'objectif de cette thèse est de définir une architecture matérielle et logicielle capable d'être déployée en minimisant sa consommation énergétique et susceptible de fournir des informations riches à l'utilisateur final.

- Au niveau matériel, nous proposons une architecture hiérarchique à plusieurs niveaux capable d'être utilisée dans différents contextes.
- Au niveau logiciel, nous introduisons deux nouveaux algorithmes de compression de données pour optimiser l'utilisation du canal de communication et une technique de stockage d'information sur support amovible capable de stocker une quantité importante de données.

Cette thèse est structurée de la manière suivante :

Le premier chapitre définit l'élément principal qui constitue un réseau de capteurs : le capteur et les différentes unités qui le composent. Cette présentation mettra l'accent sur deux unités particulières qui sont : l'unité de calcul qui est l'intelligence du dispositif et l'unité de communication qui traduit la notion de réseau proprement dite. Toutes ces unités sont génériques car elles dépendent des plateformes et des applications visées telle que l'unité de capture d'information qui est plus proche de l'électronique que de l'informatique, mais elles influent toutes sur le module de gestion énergétique. L'objectif de cette partie est de se familiariser avec l'entité élémentaire qui constitue ce type de réseau et d'identifier sa principale source de consommation qui est le module de transmission.

Le chapitre 2 illustre les différentes plateformes de réseaux de capteurs et l'évolution des projets dans lesquels elles ont été impliquées avant d'aboutir à un compromis entre les différentes plateformes qui est l'utilisation d'architecture hybrides et hiérarchiques par certains projets utilisant les réseaux de capteurs. L'aperçu des projets existants va nous permettre non seulement d'identifier les plateformes utilisées et leurs évolutions mais surtout de cerner les différents axes de recherche, les contraintes de déploiements et les améliorations qu'il reste à apporter.

Le chapitre 3 décrit notre architecture générale de réseau de capteurs hiérarchique et générique. Nous évaluons tout d'abord les performances des dispositifs de niveau haut pendant le transfert d'images à l'aide de la technologie sans fil Bluetooth vers le puits. Puis, nous étudions le compromis entre le délai et la consommation intrinsèque au standard IEEE 802.15.4 communément utilisé dans les réseaux de capteurs et qui est le moyen de communication entre les deux premiers niveaux.

Le chapitre 4 présente de nouvelles techniques de compression de données. Cette technique a pour principal objectif d'économiser l'énergie mais se révèle être aussi, dans certains cas, une aubaine pour les applications contraintes par le temps. La littérature sur les algorithmes de compression pour réseau de capteurs reste encore pauvre, mais notre travail s'est traduit par la proposition de nouvelles techniques de compression de données qui sont plus efficaces que l'algorithme de référence dans le domaine. Nos propositions sont efficaces en terme de compression, d'économie d'énergie mais aussi garantissent un temps de transfert plus rapide dans certains cas.

Le chapitre 5 décrit un nouvel environnement de stockage d'information développé sous TinyOS et utilisant le système de fichier FAT16. Nous introduisons la portabilité du système de fichier FAT (File Allocation Table) sous TinyOS-2.x pour faciliter le stockage d'une quantité importante d'information sur des supports de stockage de masse non volatile. Ce nouvel outil, générique car utilisable par tous les systèmes d'exploitation modernes, permet à un utilisateur de stocker une quantité importante d'informations lorsque les conditions d'expérimentales ne seront pas propices à l'utilisation de la radiofréquence. Ainsi, tout utilisateur pourra sans manipulation technique supplémentaire directement visualiser et analyser les données collectées sur son ordinateur. En outre, des données d'origines différentes, c'est-à-dire représentant diverses grandeurs physiques, peuvent être stockées dans des fichiers différents, ce qui facilitera l'analyse des données.

Première partie

Etat de l'art

1

Généralités sur les réseaux de capteurs

Sommaire

1.1	Le capteur	17
1.1.1	Architecture d'un capteur	17
1.1.2	TinyOS : environnement de développement dédié aux capteurs	18
1.2	Etat de l'art sur les réseaux personnels sans fil	21
1.2.1	IRDA	21
1.2.2	IEEE 802.15.1/Bluetooth	21
1.2.3	Wibree - Ultra Low Power (ULP) Bluetooth	23
1.2.4	HOME RF	24
1.2.5	IEEE 802.15.3/UWB	24
1.2.6	IEEE 802.15.4/ZigBee	24
1.3	Les réseaux <i>ad hoc</i>	27
1.3.1	Les protocoles hiérarchiques	28
1.3.2	Les protocoles géographiques	29
1.3.3	Les protocoles à plat	29
1.4	Les réseaux de capteurs sans fil	29
1.4.1	Le déploiement	30
1.4.2	La couverture de surface et la collecte de données	30
1.4.3	La contrainte énergétique	31
1.5	Les applications utilisant les réseaux de capteurs	32
1.5.1	Les applications militaires	32
1.5.2	Les applications médicales	32
1.5.3	Les applications liées à la sécurité des infrastructures	32
1.5.4	La domotique	33
1.5.5	Les applications environnementales	33
1.6	Conclusion	33

Le monde des réseaux et télécommunication est un univers en perpétuelle évolution qui se manifeste par l'enchaînement de dispositifs électroniques variés sur le marché. En effet, les progrès technologiques actuels ont favorisé l'accroissement des performances des processeurs, des mémoires et des technologies sans fils, entraînant ainsi le développement de nouveaux systèmes informatiques et électroniques. Ce développement est accéléré par l'intérêt pour des dispositifs informatiques miniaturisés, performants et autonomes par le grand public. On assiste alors à l'engouement par de grosses entreprises informatique pour la course à la miniaturisation de dispositifs électroniques embarqués tel que le témoigne la mise sur le marché de l'iPhone d'Apple¹. En effet, Apple devenue célèbre avec son architecture 8 bits, l'ordinateur personnel Apple II en 1977, puis avec la gamme Macintosh à partir de 1984, présente aujourd'hui un smartphone de dimension 115 x 61 x 11,6 mm³ ayant un processeur ARM cadencé à 620MHz. Ce bijou électronique combinant plusieurs usages à savoir : téléphone mobile, terminal Internet, assistant personnel, lecteur numérique et console de jeux, est non seulement beaucoup plus puissant que l'Apple II mais surtout beaucoup plus petit. Nous sommes passés d'un ordinateur personnel ayant une architecture 8 bits à un téléphone multi-fonctions doté d'une architecture 32 bits. Ainsi, les dispositifs embarqués d'aujourd'hui sont plus puissants que les ordinateurs d'hier.

Avec ces différents progrès remarquables et la course à la miniaturisation, nous assistons à l'émergence de systèmes embarqués combinant divers composants électroniques tels que des capteurs, des modules de communication sans fil, voire des modules de géolocalisation, capables d'observer leurs environnements, se positionner mais aussi transmettre des informations.

Par contre, la contrainte induite par l'autonomie des dispositifs embarqués incite à utiliser des plateformes ayant des performances de plus en plus réduites pour leur consommation malgré les puissances de calcul et tout cela dans un système dit collaboratif qui ne se limite plus à une seule ressource restreinte mais utilise les capacités de plusieurs autres. En effet, malgré le fait que le processeur de l'iPhone, par exemple, soit cadencé à 620MHz, pour une meilleure gestion de l'énergie, son processeur fonctionne à 412MHz. Ainsi, comme les puissances de calcul sont limitées, plusieurs nœuds s'unissent pour réaliser un travail en commun [1]. Ces techniques sont très utilisés comme le témoigne avec l'évolution d'internet le succès des systèmes pair-à-pair tels que Napster [2], E-Mule [3], BitTorrent [4] ou Skype [5] permettant de distribuer la puissance de calcul et de stockage. Nous assistons alors au développement de nouveaux types de réseaux distribués de micro-composants peu coûteux, capables de collecter des informations, les traiter puis les transmettre à l'aide d'une communication sans fil à une station de base : *les Réseaux de Capteurs sans Fil (RdCs)*.

Les RdCs vont révolutionner le paradigme de collecte et de traitement d'informations dans divers environnements. De petites tailles et auto-organisés en réseaux denses, d'où leur identification de "smart dust", c'est à dire poussière intelligente, ils sont au centre de diverses applications dans plusieurs secteurs :

- surveillance de zones sensibles pour les militaires,
- détection de feu de forêt,
- surveillance d'entrepôts chimiques,
- diagnostic médical...

¹Apple est une société multinationale américaine d'informatique devenue célèbre avec l'ordinateur personnel Apple II (1977), puis la gamme Macintosh (depuis 1984). Depuis 2001, Apple a diversifié ses activités en se focalisant aussi sur les lecteurs numériques avec l'iPod et l'iTunes Store (2003) et les téléphones portables avec l'iPhone (2007).

L'opération élémentaire au niveau de ces réseaux est la collecte systématique d'informations. Celle-ci est possible grâce aux capteurs qui informent sur leurs environnements respectifs. De ce fait, avant d'aller plus loin dans la définition des réseaux de capteurs, il est important de présenter l'entité élémentaire qui compose ce type de réseau : le capteur.

1.1 Le capteur

Traditionnellement, un capteur se définit comme un dispositif qui transforme l'état d'une grandeur physique observée en une grandeur utilisable, généralement électrique. Il constitue l'élément de base des systèmes d'acquisition de données. Plusieurs types de capteurs ont été développés dont les plus connus sont les suivants : les capteurs de distance, les capteurs tactiles, les capteurs de vision, les capteurs de température, les capteurs d'humidité...

La diversité des capteurs, comme l'illustre l'énumération ci-dessus, a permis une observation du monde et son interprétation. Cependant, avec les progrès technologiques cités plus haut sont apparus de nouveaux types de capteurs dotés d'une *intelligence*. Cette intelligence se traduit par l'utilisation d'une unité de calcul capable de réaliser des traitements sur les données capturées.

1.1.1 Architecture d'un capteur

La Fig. 1.1 illustre en détail les quatre principales unités qui composent un capteur [6] dit *intelligent* :

- l'unité de capture d'informations : elle est composée du capteur proprement dit et de l'ADC² qui transforme les signaux analogiques en signaux numériques. En effet, le capteur observe un phénomène et fournit les signaux analogiques correspondant au phénomène au convertisseur analogique/numérique, l'ADC, qui les transforme en signaux numériques compréhensible par l'unité de traitement. L'ADC est un exemple possible mais il existe aussi les accéléromètres qui fonctionnent en PWM³ et effectuent une mise en forme du signal pour le traitement.
- l'unité de traitement : cette unité constitue l'élément central du capteur. Elle peut être couplée à une unité de stockage et contrôle les procédures qui assurent la collaboration avec les autres unités. Elle peut fonctionner avec des systèmes d'exploitation spécialement conçus pour les capteurs dont le plus connu est TinyOS [7], jusqu'à Linux pour des nœuds plus complexes qui vont aussi nous intéresser.
- l'unité de communication : appelée aussi *transceiver*, néologisme anglo-saxon issu de la contraction de transmitter et receiver, représente le couple émetteur/récepteur qui connecte le nœud au réseau. Elle est associée à une technologie de transmission d'information telles que la radiofréquence, l'optique etc.
- l'unité d'alimentation : elle fournit l'énergie nécessaire au fonctionnement du dispositif. Cependant, les dispositifs embarqués étant de petites tailles, les batteries sont aussi petites et donc les ressources énergétiques sont limitées. Cette limitation fait que la gestion énergétique est un point critique pour le capteur et ainsi pour les réseaux de capteurs. Il faut aussi noter que la taille des batteries facilite le transport et fait que ces systèmes sont de plus en plus mobiles.

²ADC est un acronyme d' Analog-to-Digital Converter. C'est un convertisseur analogique-numérique.

³PWM est un acronyme de Pulse Width Modulation dont la traduction en français est : Modulation de Largeur d'Impulsions (MLI).

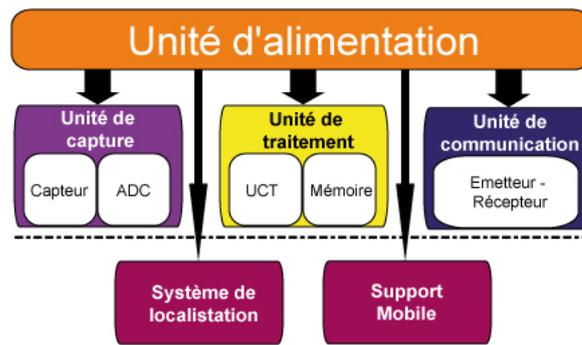


FIG. 1.1 – Les composants d'un capteur

La Fig. 1.1 a été conçue avec un ordre de priorité du haut vers le bas. En effet, le point critique au niveau de cette thématique est la gestion d'énergie. C'est pour cette raison que le module d'alimentation est le premier en allant du haut vers le bas. Plusieurs thématiques de recherche dans ce domaine gravitent autour de la gestion d'énergie dans les réseaux de capteurs car la durée de vie du réseau, de son nom anglais *lifetime*, qui est le temps pendant lequel un réseau de capteurs est susceptible d'être efficace et capable d'observer un environnement, dépend de la durée de vie de l'entité élémentaire qui le compose c'est-à-dire le capteur.

D'autres modules additionnels qui se trouvent à la base de notre figure, relativement moins pertinents et surtout très gourmands en énergie sont : un système de géolocalisation ou un support mobile. En effet, dans un réseau statique où la position géographique des capteurs est une information pertinente, comme par exemple pour la détection de feux de forêt où il faut situer les capteurs, il est indispensable d'avoir un système de géolocalisation comme un récepteur GPS par exemple ou encore disposer d'un support mobile dans un réseau mobile. Cependant, il faut noter que disposer des coordonnées d'un nœud dans un réseau à forte mobilité est très coûteux énergétiquement car la position du capteur est recalculée régulièrement.

Pour faire fonctionner un capteur, l'unité de traitement qui est l'organe essentiel, peut disposer d'un système d'exploitation. Par contre, la grande question est de savoir quelles doivent être les principales caractéristiques d'un système d'exploitation pour RdC.

1.1.2 TinyOS : environnement de développement dédié aux capteurs

Un système d'exploitation, *Operating System* en anglais et abrégé OS, se définit comme un programme garantissant la liaison entre les ressources matérielles et les logiciels (applications informatiques). Pour faire fonctionner un capteur, un système d'exploitation peut constituer un outil intéressant, cependant, soumis aux contraintes sévères des capteurs, il est spécifique. En effet, les systèmes d'exploitation classiques utilisés par les ordinateurs et les plateformes disposant de ressources importantes ne sont pas adaptés aux capteurs car ils offrent un support natif pour un grand nombre de périphériques, ce qui demande des ressources conséquentes que les capteurs n'ont pas forcément. Mais ce défaut est aussi un avantage car l'architecture qui est mise en œuvre permet à une application d'être en mesure de pouvoir fonctionner sur n'importe quelle machine sans pour autant connaître les caractéristiques matérielles, donc bas niveau de celle-ci.

Par contre, l'informatique embarquée se base historiquement sur des langages de bas niveau

tels que le C ou l'assembleur. Le programme dialogue directement avec le matériel, ce qui le rend plus léger en terme de ressources et de capacités. En effet, il n'est plus sur un modèle de plus grand dénominateur commun comme l'OS mais sur un modèle de spécificité qui lui apporte cette caractéristique. Cependant, cet avantage est aussi son inconvénient. Le logiciel et le matériel étant fortement liés, toute modification matérielle entraîne nécessairement une modification logicielle. De là, on peut noter qu'un système embarqué impose une solution pointue et économe en terme de ressources qui rend l'utilisation des systèmes d'exploitation classiques difficile. De ce fait, différents OS ont été développés pour systèmes embarqués et aussi plus spécialisés pour réseaux de capteurs [8] tels que :

- **Contiki** [9] : est un système d'exploitation *open source*, léger, multitâche, générique développé pour tout système embarqué ayant des contraintes mémoire. La portabilité de ce système va de l'ordinateur ayant une architecture 8 bits aux systèmes embarqués sur des microcontrôleurs comme certaines plateformes pour RdC.
- **FreeRTOS** : est un noyau de système d'exploitation temps réel pour système embarqué et portable sur plusieurs microcontrôleurs.
- **TinyOS** : est le premier système d'exploitation spécialement dédié aux réseaux de capteurs. C'est un OS *open source* initialement conçu à l'Université de Berkeley en Californie et qui connaît une notoriété dans le domaine des réseaux de capteurs non seulement parce qu'il est le premier mais aussi parce qu'il est utilisé par les plateformes CROSSBOW [10], les plus populaires dans le domaine qui ont été aussi développés par la même université. Le langage utilisé par TinyOS est nesC [11] qui est une extension du langage C.
- **Mantis OS** [12] : est aussi un OS pour réseaux de capteurs entièrement écrit en C et développé par une équipe de l'université du Colorado. Ce système utilise une approche différente des autres OS pour réseaux de capteurs sans fil qui est *thread driven execution model* différent de TinyOS qui utilise un *event driven model*.
- **SOS** [13] : est un OS pour réseaux de capteurs utilisant le même modèle, *non-preemptive event driven scheduler*, que TinyOS mais écrit en C. Il a été développé à l'université de Californie, Los Angeles (UCLA).
- **NutOS** : c'est un OS temps réel open source conçu pour système embarqué. Il est multitâche et offre une pile TCP/IP. Il a constitué une base pour ETHERNUT [14] ou encore pour l'OS BTNut utilisé dans les BTnodes [15] et où une pile Bluetooth open source a été rajoutée.

Les OS qui précèdent sont spécialement dédiés aux plateformes de réseau de capteurs qui ont des ressources limitées. Aussi, il faut noter qu'il existe d'autres plateformes de haut niveau avec plus de ressources comme celle que nous allons utiliser dans ce travail : la carte Fox [16] qui tourne sur un système d'exploitation GNU/Linux qui offre plus de fonctionnalités.

Après avoir énuméré les différents OS pour réseaux de capteurs, nous allons présenter le plus populaire d'entre eux, qui est également celui que nous avons utilisé.

TinyOS

Contrairement aux systèmes d'exploitation classiques, TinyOS se présente comme une solution intermédiaire entre une application bas-niveau apportant une faible empreinte mémoire mais trop spécifique au matériel et un système d'exploitation apportant une réelle notion d'abstraction entre l'application et le matériel mais qui n'est pas prévu pour du matériel à faible ressources.

TinyOS n'est pas un système d'exploitation au sens actuel du terme. Il n'offre pas par exemple de notion de multitâches, d'utilisateurs ou de système de fichier. Il n'y a pas de notions de mode utilisateurs et de mode noyau. En fait, TinyOS met un ensemble d'outils à la disposition du programmeur en vue de simplifier le développement. Il utilise une extension du langage C pour systèmes embarqués appelée NesC [17] et l'exécutable produit est en tout point semblable à ce que pourrait générer une application bas niveau.

L'intérêt de TinyOS ne réside pas spécialement dans l'application installée sur le capteur mais au niveau de la simplification du développement. Il apporte la même notion d'abstraction qu'un système d'exploitation classique, sans imposer une trop forte dépendance entre le matériel et le logiciel rendant dès lors possible une portabilité quasi transparente sur une multiplicité de plateformes différentes, et surtout sans aucunes contraintes fortes liées au matériel.

TinyOS a été spécifiquement conçu pour les réseaux de capteurs [18] et donc répond à leurs exigences en terme de ressources. Il utilise un fonctionnement évènementiel différent du fonctionnement dit temporel où les actions du système sont gérées par une horloge donnée.

L'architecture de TinyOS est basée sur une association de composants, ce qui a réduit la taille du code nécessaire à sa mise en place. Une application écrite dans ce langage se compose d'un ou plusieurs composants assemblés. Les composants ont deux champs principaux : un pour leur spécification qui contient les noms de leurs interfaces, et un deuxième pour l'implémentation. Un composant fournit et utilise des interfaces.

Les interfaces sont fournies dans le but de représenter les fonctionnalités qu'offre le composant à son utilisateur dans sa spécification. Il existe deux types de composants dans nesC : module et configuration. Les modules fournissent les implémentations d'une ou de plusieurs interfaces. Les configurations sont utilisées pour l'assemblage d'autres composants ensemble, en connectant les interfaces utilisées par les composants pour les interfaces fournies par d'autres.

Après avoir défini un outil permettant à l'unité de traitement de fonctionner, nous allons nous intéresser à une autre unité tout aussi importante. En considérant, les quatres unités de base définies plus haut, la principale source de consommation énergétique est l'unité de communication. Approximativement 80% de l'énergie consommée par un nœud, l'est pendant la transmission d'informations [19].

Aussi, cette unité est associée à une technologie sans fil donc le choix d'une technologie sans fil est très important et exige de trouver un bon compromis entre la consommation, le débit et la portée en fonction de l'application visée. Il existe différentes technologies sans fil mais les plus adaptées aux systèmes embarqués et plus spécialement aux réseaux de capteurs sont celles utilisées par les réseaux de proximité sans fil pour leur faible consommation. C'est pour cette raison que nous allons faire un état de l'art sur les réseaux personnels sans fil [20].

1.2 Etat de l'art sur les réseaux personnels sans fil

Contrairement aux réseaux locaux notés WLAN⁴ tels que Wifi (IEEE 802.11x) et ses multiples standards⁵ ou encore hiperLAN2 (HIGH Performance Radio LAN 2.0) dont l'objectif est de permettre une communication sans fil avec un rayon d'une centaine de mètres et qui consomment beaucoup d'énergie de l'ordre du Watt, les réseaux personnels sans fil ou réseaux sans fil de proximité, notés WPANs⁶, concernent les réseaux à faible portée de l'ordre de quelques mètres voire dizaines de mètres. Ils sont utilisés pour connecter des dispositifs autonomes entre eux et donc qui peuvent s'adapter aux réseaux de capteurs.

1.2.1 IRDA

L'IRDA qui est un acronyme de *Infrared Data Association*, communément appelé *infrarouge*, est un moyen de communication qui utilise comme médium la lumière infrarouge pour transmettre des informations. Le standard initial, normalisé en 1994, offrait un débit de 115 Kbit/s qui a abouti en 1999 à une extension allant jusqu'à 16 Mbits/s. La principale caractéristique de l'IRDA qui est aussi son principal inconvénient est que deux périphériques utilisant cette technologie doivent être en ligne de vue pour pouvoir communiquer. En effet, les transmetteurs IRDA étant directifs avec un angle d'environ 15°, ils doivent être bien orientés pour communiquer. Cette contrainte de fonctionnement pour cette technologie qui a une faible consommation électrique fait qu'au niveau sécurité elle limite les possibilités d'interception du signal. Au début des années 1990, l'IRDA était très utilisé pour les téléphones portables, les ordinateurs et d'autres périphériques mais avec l'apparition de Bluetooth, la donne a changé.

1.2.2 IEEE 802.15.1/Bluetooth

Ce standard de communication créé en 1994 par Ericsson, a été défini à la base pour remplacer les câbles et combler les lacunes de l'IRDA qui était populaire à l'époque. C'est une technologie radio courte distance, faible consommation, basée sur des puces électroniques peu coûteuses et destinées à simplifier les connexions entre appareils électroniques. En 1998, d'autres entreprises rejoignent Ericsson pour former le Bluetooth Special Interest Group (SIG). Cependant, Bluetooth n'a pas seulement pour but de faire coopérer des périphériques de constructeurs différents en définissant un système de communication par ondes hertziennes sur la bande ISM⁷ de 2,4 GHz mais définit réellement une pile logicielle complète, contrairement à WiFi qui n'offre que le niveau 1 et le niveau 2 de la pile OSI. Elle permet aux périphériques de se découvrir et de communiquer entre eux sans savoir quels services ils offrent à la base. On distingue 3 classes de modules radio Bluetooth sur le marché ayant des puissances différentes et donc des portées différentes :

Classe	Puissance	Portée
1	100mW(20dBm)	100 mètres
2	2,5mW(4dBm)	15 à 20 mètres
3	1mW(0dBm)	1 mètres

La plupart des fabricants d'appareils utilisent des modules de classe 2 pour une consommation énergétique moindre, de l'ordre de 320 mW en fonctionnement et 60 mW hors association avec

⁴Wireless Local Area Network

⁵Les différents standards WiFi : a, b g, n et y diffèrent par la portée de leur signal et leur taux de transfert.

⁶Wireless Personal Area Networks

⁷ISM est un acronyme de Industrial, Scientific, Medical

d'autres périphériques, et un rayon de communication compatible avec les applications visées en domotique par exemple.

Les fonctionnalités des différentes couches de la pile Bluetooth (Fig. 1.2) sont les suivantes :

- L'interface de contrôle *HCI* (*Host Control interface*) propose une méthode d'accès uniforme aux fonctionnalités de la couche *Baseband* indépendamment de l'interface d'accès du matériel Bluetooth.
- Le protocole *L2CAP* (*Logical Link Control and Adaptation Protocol*) assure la transmission asynchrone des paquets. Il utilise le multiplexage, la segmentation et le réassemblage.
- Le protocole de découverte de service *SDP* (*Service Discovery Protocol*) permet à un appareil Bluetooth de rechercher d'autres appareils et d'identifier les services disponibles.
- Le protocole *RFCOMM* sert à transmettre des données aux couches de haut niveau. C'est un service basé sur les spécifications RS-232, qui émule des liaisons séries et peut notamment servir à faire passer une communication IP par Bluetooth.
- Le protocole *OBEX* (*Object Exchange*) est un protocole d'échange développé à la base pour l'IRDA permettant de transférer des objets .

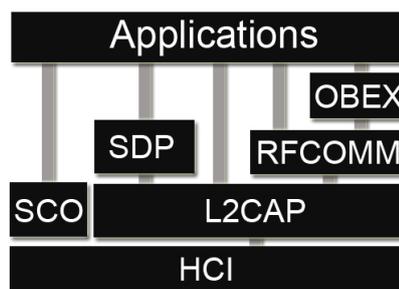


FIG. 1.2 – Pile Bluetooth

Une spécificité de Bluetooth est le type de topologies qu'il offre. Les périphériques Bluetooth qui utilisent un même canal de communication forment un *Piconet*, constitué d'un maître et jusqu'à 255 esclaves mais seulement 7 esclaves peuvent être actifs, les autres sont "parqués" (Fig. 1.3).

Deux types de liens peuvent être établis avec un maître et un ou plusieurs esclaves : des liens synchrones orientés connexion (SCO⁸), souvent utilisés pour le transport de la voix, et des liens asynchrones sans connexion (ACL⁹).

L'interconnexion de plusieurs Piconets, appelé *scatternet* (Fig. 1.3), se fait au moyen d'un nœud qui est soit esclave dans un *Piconet* et maître dans l'autre *Piconet* soit esclave dans les deux *Piconets* car un nœud ne peut être maître dans différents *Piconets* simultanément. La formation des *scatternets* constitue un axe de recherche important dans le domaine des réseaux sans fil qui utilise la technologie Bluetooth [21, 22, 23]. En effet, la scalabilité d'un réseau Bluetooth est un vrai handicap lorsque l'on veut étendre cette technologie à des réseaux denses qui s'éloignent d'une utilisation domestique.

⁸Synchronous Connection-Oriented

⁹Asynchronous Connection-Less

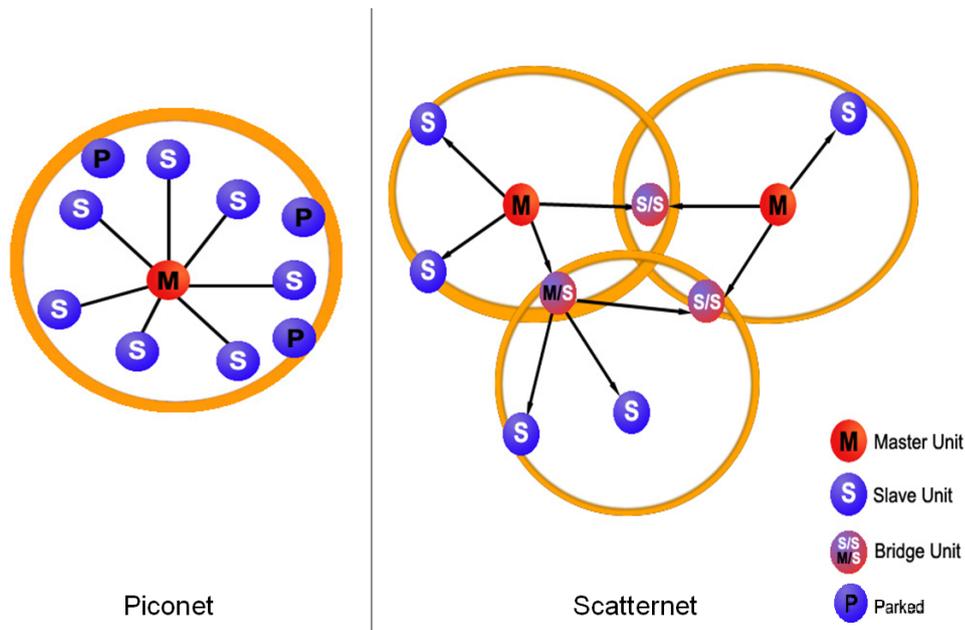


FIG. 1.3 – Représentation des différents modèles réseau de Bluetooth

La bande de fréquence de 2.4GHz qu'utilise Bluetooth, libérée dans la plupart des pays du monde, est encombrée par plusieurs autres protocoles (WiFi, ZigBee, ...) et polluée par les fours à micro-ondes.

C'est pour cette raison que pour éviter les interférences avec d'autres modules radios, bluetooth utilise la technique FHSS (*Frequency Hopping Spread Spectrum*) qui est un étalement de spectre par saut de fréquence. Cette technique consiste à découper la bande de fréquence en 79 canaux d'une largeur de 1 MHz et change aléatoirement et fréquemment de fréquence avec une période de $625 \mu s$, ce qui permet 1600 sauts par seconde.

Par contre, la mobilité, ou encore la transmission à travers des éléments non métalliques sont garanties par Bluetooth au détriment de la sécurité contrairement à l'IRDA.

1.2.3 Wibree - Ultra Low Power (ULP) Bluetooth

Le succès de Bluetooth fait qu'une nouvelle norme conçue par Nokia a été annoncée en Octobre 2006. Cette norme s'appelle Wibree [24] devenue Ultra Low Power Bluetooth car le Bluetooth SIG l'a acceptée en accord avec Nokia comme variante de la norme Bluetooth. Wibree est une technologie sans fil qui offre un débit d'environ 1 Mbps dans un rayon d'une dizaine de mètres et ne se positionne pas comme un concurrent de Bluetooth mais plutôt comme un complément. Elle opère dans la bande de 2,4 GHz et son principal avantage est sa consommation qui promet d'être 10 fois moins gourmande en énergie que Bluetooth. Cette norme constituera un bon compromis entre le débit et la consommation énergétique. On pourra ainsi l'adapter à des dispositifs de petite taille comme des montres par exemple.

1.2.4 HOME RF

HomeRF [25] est une spécification de réseau sans fil conçue comme son nom l'indique pour les réseaux domestiques mais qui n'a pas connu un grand succès. Cette technologie permettrait aussi entre autre de soutenir la technologie de transport de la voix en mode numérique sur les réseaux sans fil appelée DECT qui est un acronyme de *Digital Enhanced Cordless Telephone*. Elle a été imaginée par un groupe de sociétés appelé HomeRF Working Group dont les principaux acteurs étaient Compaq, HP, IBM, Intel et Microsoft. Cependant, la disponibilité de la norme WiFi pour des réseaux domestiques, a entraîné l'éloignement de deux gros sponsors, Intel et Microsoft, ce qui fait que cette norme était en perte de vitesse. Malgré son accès au médium intéressant, *Shared Wireless Access Protocol*, qui est un mélange de CSMA/CA¹⁰ pour les données asynchrones et TDMA¹¹ pour les données isochrones, elle est abandonnée aujourd'hui.

1.2.5 IEEE 802.15.3/UWB

L'UWB, qui est un acronyme de *Ultra Wide Band* ou encore *Ultra large bande passante en français*, est une technologie sans fil conçue pour faire communiquer des périphériques grand public sur une courte distance avec un haut débit, tout en considérant une des principales contraintes des dispositifs de domotique qui est l'autonomie. En effet, cette technologie, qui consomme peu d'énergie, est basée sur une technique d'étalement de spectre permettant de transmettre des données sur un très large spectre en un temps très court. Ce profil fait qu'elle est bien adaptée pour le transport de données nécessitant un haut débit tel que le multimédia. La promotion de cette technologie est assurée par un groupement d'industriels appelé *WiMedia Alliance*. Cependant, le coût des composants et la concurrence d'autres technologies comme l'USB sans fil ou encore Bluetooth fait qu'elle est en perte de vitesse. Cela se traduit par le départ d'un des promoteurs, Intel en novembre 2008 et l'arrêt en Mars 2009 de WiMedia. Néanmoins, il faut noter que les briques de l'UWB se retrouveront dans les spécifications des normes de l'USB sans fil et Bluetooth 3.0. En effet, la norme Bluetooth 3.0. encore appelée *Next-Gen Bluetooth ou Bluetooth UWB* aura sa couche matérielle basée sur l'UWB permettant d'atteindre un débit de 480 Mbits/s.

Face à ces différentes technologies, une s'est démarquée considérablement par rapport aux autres pour les réseaux de capteurs : IEEE 802.15.4/ZigBee.

1.2.6 IEEE 802.15.4/ZigBee

Contrairement à plusieurs applications et à plusieurs technologies sans fil comme celles qui précèdent où le débit est une priorité, l'émergence d'un nouveau type d'applications contraintes par la consommation énergétique pour une meilleure autonomie des dispositifs utilisés dans ce type de réseau a donné lieu à un nouveau standard à faible débit et à consommation réduite : IEEE 802.15.4 [26].

Ce nouveau standard, qui fait partie du groupe des *Low-Rate Wireless Personal Area Networks (LR-WPANs)* c'est-à-dire *aux réseaux personnels sans fils à bas débit*, a été conçu pour

¹⁰CSMA est un acronyme de Carrier Sense Multiple Access with Collision Avoidance. C'est une méthode d'accès au médium qui utilise un mécanisme permettant d'éviter des collisions à l'aide d'un principe d'accusé de réception réciproque entre l'émetteur et le récepteur.

¹¹TDMA est un acronyme de Time Division Multiple Access. C'est un mode de multiplexage temporel permettant de transmettre plusieurs signaux sur un même canal en découpant le temps disponible entre différentes connexions.

connecter les dispositifs sans batterie ou contraint par leur batterie limitée en énergie. Son objectif est donc de consommer peu d'énergie de sorte qu'une batterie puisse tenir pendant très longtemps.

Ce standard spécifie les couches basses, MAC¹² et physique, pour les LR-WPANs. Comme l'IEEE ne définit que la couche MAC et la couche physique, un groupe d'entreprises appelé la ZigBee Alliance [27] a spécifié les couches hautes, c'est-à-dire la couche réseau et la couche application pour ce standard allant du routage à l'application, ce qui a donné naissance au protocole ZigBee (Fig. 1.4). Un faible débit, une faible portée et une faible consommation énergétique sont les principales caractéristiques d'un RdC classique : c'est pour cette raison que dans notre étude, nous allons nous intéresser à cette technologie sans fil qui utilise, comme Bluetooth et WIFI, la bande ISM de 2,4 GHz et offre un débit de 250 Kbit/s. Cette technologie fera l'objet d'une étude plus complète dans les chapitres suivants.

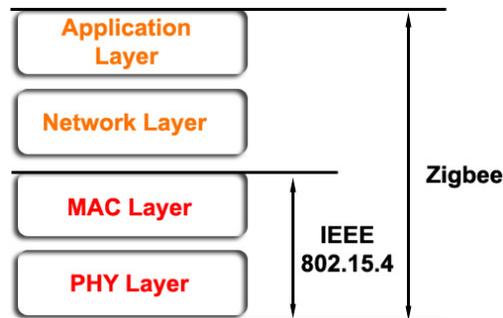


FIG. 1.4 – La pile du protocole IEEE 802.15.4/ZigBee

D'une manière générale, nous pouvons conclure que les protocoles de communication changent régulièrement avec une durée de vie inférieure à 10 ans. Cela montre que ce domaine n'est pas encore assez mature et rend le travail du chercheur, dépendant des industriels du domaine, difficile.

Après s'être principalement focalisé sur deux des quatre principales unités d'un capteur qui sont : l'unité calcul et son système d'exploitation puis les technologies sans fil susceptibles d'être utilisées par l'unité de transmission (*ou transceiver*), nous allons observer la Fig. 1.5.

Cette figure montre que pour une unité de calcul précise, c'est à dire pour un micro-processeur donné, l'unité de communication correspondante, c'est à dire le transceiver ayant en moyenne le même débit, consomme plus d'énergie. Cette image montre aussi les technologies sans fil susceptibles d'être adaptées pour les réseaux de capteurs de par leur consommation énergétique telles que : ZigBee (IEEE 802.15.4), Bluetooth (IEEE 802.15.1) ou encore Wibree voir même WiFi (IEEE 802.11x) et ses multiples standards.

ZigBee et Bluetooth sont plus souvent utilisés à cause de leur faible consommation mais leur débit et la portée de leur signal est limité. Aussi, l'architecture complexe de Bluetooth et ses communications point à point font que Zigbee, facilitant les communications "un-vers-tous" pour le passage à l'échelle, connaît un plus grand succès dans les réseaux de capteurs.

¹²Medium Access Control

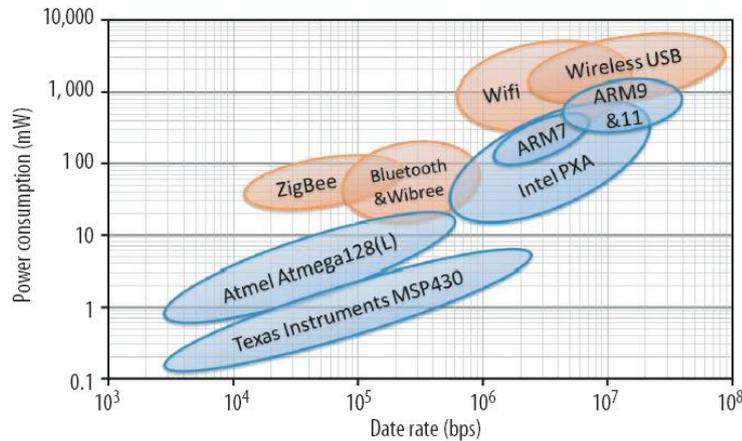


FIG. 1.5 – Comparaison énergétique entre transceiver et micro-processeurs [28]

C'est la mise en réseau de plusieurs de ces capteurs dit *intelligents* et leur collaboration à l'aide de leur module de communication qui forment un RdC. Les capteurs deviennent ainsi des entités capables d'opérer en toute autonomie, dans tout environnement même hostile à l'homme, afin de collecter, traiter et envoyer des données relatives à un endroit du globe bien précis vers un lieu d'analyse, une station puits appelée *Sink*. Cette station transmet ensuite les données via un réseau classique, Internet ou par un canal de communication à haut débit qui est chargé de les analyser afin de prendre des décisions (Fig. 1.6).

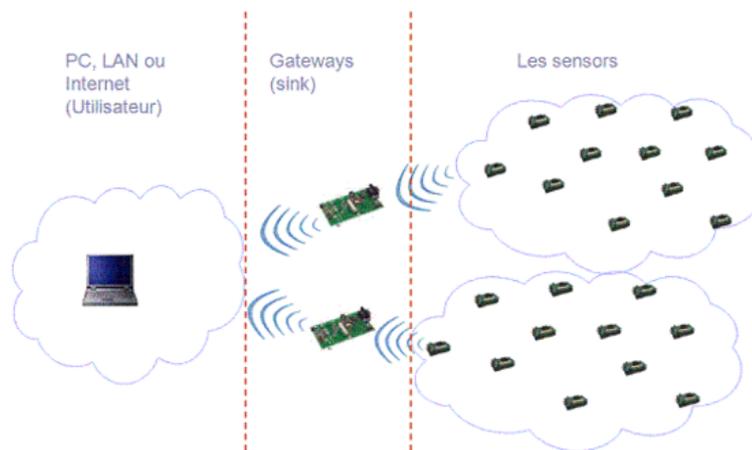


FIG. 1.6 – Architecture générale d'un réseau de capteurs sans fil [29]

Pour finir, nous allons introduire deux problématiques auxquelles nous allons nous intéresser tout au long de cette thèse et qui sont illustrées sur la Fig. 1.5 :

- la contrainte énergétique, telle que décrite sur la Fig. 1.5, pousse à se focaliser sur des plateformes ayant des microcontrôleurs de type TI MSP430 ou Atmel Atmega128 à fréquence limitée, en utilisant la technologie ZigBee à débit limité, et qui ont une consommation de l'ordre du μW en mode veille. Ce constat est confirmé par la multiplicité de plateformes de réseaux de capteurs utilisant ces microcontrôleurs et cette technologie sans fil, ZigBee, basée sur le standard IEEE 802.15.4.

Cependant, nous pouvons nous interroger sur la possibilité de réaliser des applications complexes dans les réseaux de capteurs, tel que le multimédia de haute qualité par exemple, à l'aide de ces dispositifs. En effet, le multimedia peut constituer une information supplémentaire riche pour l'utilisateur final d'un réseau de capteurs.

Actuellement, il est difficile de faire du multimédia dans les réseaux de capteurs qui utilisent une technologie sans fil à bas débit. Les plateformes actuelles de réseau de capteurs développées pour être économe en énergie ont des puissances de calcul limitées et utilisent la technologie sans fil ZigBee. Ce choix ne limite-t-il pas le champ d'application des réseaux de capteurs ? Nous donnons une partie de réponse à cette problématique dans le Chapitre 3.

- la contrainte de la communication (Fig. 1.5) qui est la principale source de consommation énergétique introduit la seconde problématique : comment optimiser l'exploitation de la bande passante pour les réseaux de capteurs ? Une réponse se situe dans le Chapitre 4.

La définition d'un capteur, puis des réseaux de capteurs pousse à s'apesantir sur les environnements et les techniques qui ont servi de base à ces réseaux émergents.

1.3 Les réseaux *ad hoc*

Les réseaux de capteurs constituant une spécialisation des réseaux *ad hoc*, nous allons brièvement présenter ce type d'architecture réseau.

Originaire du latin, le mot *ad hoc* signifie *qui va vers ce vers quoi il doit aller*. Un réseau *ad hoc* se définit alors comme un système formé de nœuds ne disposant d'aucune infrastructure, communiquant par ondes radio, où chaque nœud offre un service relais permettant de réémettre un message dont il n'est pas destinataire à un nœud qui est hors de portée radio de l'émetteur du message [30]. Cette technique de réémission très utilisée dans les réseaux *ad hoc* dépourvus d'infrastructure porte le nom de *multisaut* ou *multihop*.

A la base, les réseaux *ad hoc* sont une application militaire née dans les années 1970 avec le Projet PRNet [31] de la DARPA¹³ qui correspond à l'agence pour les projets de recherche avancée de défense des Etats-Unis. Ces réseaux illustrent proprement le concept de *lieu de combat* où il n'y a pas d'infrastructure pré-existante et dans lequel le multisaut permet de pallier le problème de dispositif n'étant pas à portée radio. D'autres projets militaires sponsorisés par DARPA suivront tel que SURAN (SURvivable Radio Network) en 1983, prolongement de PRNet, ayant pour but de combler les lacunes rencontrées expérimentalement par PRnet avec un champ d'application un peu plus large que le champ de bataille ou encore GLoMo (Global Mobile Information System) en 1994 avec pour mission l'étude des possibilités d'adaptation des concepts d'internet à des terminaux mobiles.

Cette technologie finira par aussi susciter un intérêt pour les applications civiles qui se traduira par la création du groupe MANET¹⁴ par l'IETF¹⁵ en 1995. Les caractéristiques de ces réseaux font que la manière d'acheminer les informations est très importante. On distingue alors différents types de protocoles de routage illustrés sur la Fig. 1.7. Cette figure n'est qu'un support mais

¹³DARPA est un acronyme de *Defense Advanced Research Agency*

¹⁴MANET est un acronyme de *Mobile Ad Hoc Networks*

¹⁵IETF est un acronyme de *Internet Engineering Task Force*

d'autres familles et références peuvent bien sûr être ajoutées.

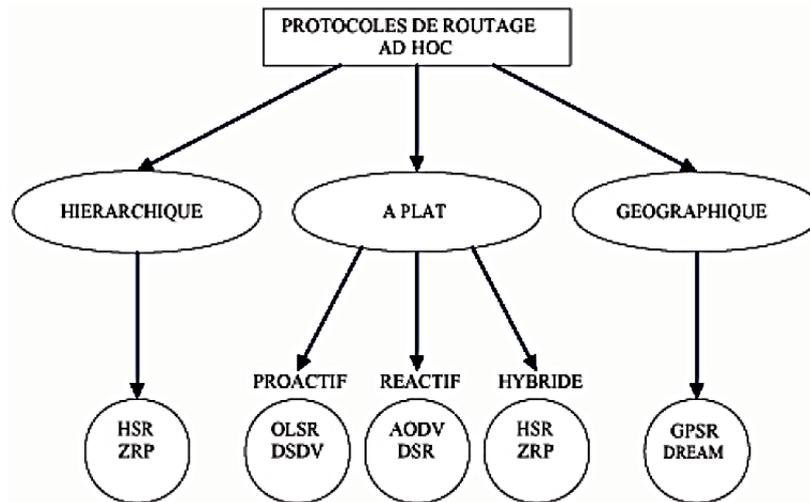


FIG. 1.7 – Classifications des protocoles dans les réseaux *ad hoc*

Les principales caractéristiques dans ces environnements sont :

- une topologie dynamique,
- une sécurité des communications limitée,
- des contraintes d'énergie,
- une bande passante limitée,
- l'absence d'infrastructure.

Sur la Fig. 1.7, les principales familles de protocoles de routage pour les réseaux *ad hoc* sont : hiérarchiques, géographiques et les protocoles à plat que nous allons respectivement présenter dans les prochains paragraphes.

1.3.1 Les protocoles hiérarchiques

Le point clé dans un protocole de routage hiérarchique est la différenciation entre les nœuds du réseau. Cette distinction peut se faire au niveau physique ou au niveau logique.

1. Au niveau physique : lorsqu'il y a une différenciation au niveau matériel, les nœuds sont hétérogènes et diffèrent de par leur capacités et fonctionnalités. De ce fait, certains nœuds disposent de caractéristiques supplémentaires et peuvent réaliser des tâches plus complexes que d'autres nœuds. Cependant, il existe un coût supplémentaire au niveau du prix et de la consommation énergétique pour les dispositifs qui ont plus de fonctionnalités.
2. Au niveau logique : dans un réseau homogène les nœuds peuvent également être hiérarchisés selon leurs fonctionnalités. Dans un découpage en cluster par exemple, permettant de partitionner le réseau et donc de le structurer, certains nœuds appelés cluster heads ou encore tête de clusters ont pour rôle d'organiser leurs clusters respectifs. Les cluster heads qui sont des nœuds de niveau supérieur dans la hiérarchisation peuvent communiquer exclusivement entre eux et forment un espace appelé *backbone*.

Le principe de base d'un protocole de routage hiérarchique est de créer une topologie virtuelle dans un réseau dense, puis de l'exploiter pour router les paquets [32, 33, 34]. Cette architecture facilite l'échange de paquets lors du passage à l'échelle de ce type de réseau car il y a des échanges intra-zones et inter-zones organisés.

1.3.2 Les protocoles géographiques

Dans les protocoles qui utilisent le concept de routage géographique, les nœuds sont supposés connaître leur position :

- Soit en utilisant un module externe comme le GPS¹⁶,
- Soit en utilisant des mesures du signal (RSSI¹⁷, TDOA¹⁸, AOA¹⁹).

Une fois les positions connues par les nœuds, ils déterminent le chemin vers une destination en fonction de ses coordonnées [35]. Le protocole LAR [36] par exemple exploite la connaissance des coordonnées du nœud émetteur et du nœud récepteur pour optimiser la procédure d'inondation. Il est aussi important de noter que l'ajout d'un module de géolocalisation, comme le GPS par exemple, est coûteux, ce qui pousse certaines solutions à se servir juste de quelques nœuds repères, appelés ancres, qui connaissent leurs positions pour déterminer la position des autres nœuds en utilisant des techniques de triangulation.

1.3.3 Les protocoles à plat

Les protocoles à plat sont les plus répandus car ils peuvent être utilisés par les autres familles de protocoles comme les protocoles hiérarchiques et géographiques. Dans ces protocoles, tous les nœuds ont le même rôle et sont homogènes en terme de ressources. On distingue trois catégories de routage pour ces protocoles :

1. le routage proactif où les routes sont établies à l'avance [37, 38, 39],
2. le routage réactif où les routes sont recherchées à la demande [40, 41],
3. le routage hybride qui constitue un mélange des deux précédents [32].

La liste des protocoles n'est pas exhaustive car ce n'est pas le sujet de cette thèse mais nous avons donné juste un aperçu de ce qui existe avec les principaux protocoles de chacune des catégories.

1.4 Les réseaux de capteurs sans fil

Nous définissons alors les réseaux de capteurs sans fil comme plusieurs nœuds, dits capteurs, déployés de façon à couvrir un territoire donné qui communiquent par radio pour concentrer l'information vers une station puits appelée Sink situé en plein cœur ou en bordure du territoire. Il est important de noter ici que l'utilisation du mot capteur est un peu *abusive* car elle ne définit pas seulement le dispositif de captage mais toute la plateforme qui forme le capteur dit intelligent. Mais dans cette thématique qui est la nôtre cet abus est toléré. La durée de vie de ces réseaux est le temps au bout duquel des données peuvent être collectées par les capteurs et transmises jusqu'à la station de base.

¹⁶Global Positioning System

¹⁷RSSI est un acronyme de Received Signal Strength Indication

¹⁸TDOA est un acronyme de Time Difference Of Arrival

¹⁹AOA est un acronyme de Angle Of Arrival

De là, nous constatons bien que les RdCs forment un nouveau type de réseaux héritier des réseaux *ad hoc* mais soumis aux contraintes sévères des capteurs telle que la gestion d'énergie. Tout au long de ce document, nous nous focaliserons sur l'élément particulier qui est le capteur et nous généraliserons à la notion même de réseau de capteurs.

Au vu de ce qui précède, nous allons présenter les réseaux de capteurs tout en dénotant les points communs et les différences avec les réseaux *ad hoc*. En effet, les réseaux *ad hoc* et les réseaux de capteurs partagent des aspects communs et se différencient par des caractéristiques qui leur sont propres :

- **Les différences** : une des principales différences est le coût énergétique car les réseaux *ad hoc* demandent plus de puissance énergétique dû au fait que les équipements qui constituent ces réseaux utilisent des modules plus puissants tels que des modules de communication qui ont des portées plus grandes et donc des puissances émises plus élevées, ce qui consomme plus d'énergie.

De ce fait, ils offrent des débits et des ressources plus élevés. Aussi, cette contrainte fait que les protocoles de routage doivent être adaptés, voire modifiés, pour les réseaux de capteurs. A cela s'ajoute une architecture possible et fréquente dans les réseaux de capteurs qui nécessite l'utilisation d'une station collectrice vers laquelle toutes les données sont rattachées pour un futur traitement tandis que pour les réseaux *ad hoc*, par définition, aucune infrastructure n'est utilisée.

- **Les points communs** : l'auto-organisation, le multi-hop et le besoin d'efficacité énergétique constituent les trois principaux points communs entre ces réseaux.

Après avoir identifié les principales caractéristiques d'un RdC par rapport à un réseau *ad hoc*, nous allons nous intéresser aux différentes étapes de la mise en place d'un réseau de capteurs.

1.4.1 Le déploiement

Le déploiement constitue la première étape de la mise en place d'un réseau de capteurs. C'est une étape, tout comme l'analyse des informations collectées par la station de base, qui nécessite une intervention humaine. Ce déploiement peut se faire de deux manières :

- déterministe : c'est-à-dire que la position des capteurs est bien déterminée et donc connue à l'avance,
- aléatoire : dans ce cas les capteurs s'auto-organisent pour former le réseau.

Les capteurs sont déployés dans le but d'acquérir des informations sur une zone bien précise d'où la nécessité de présenter les caractéristiques de la couverture de surface.

1.4.2 La couverture de surface et la collecte de données

La couverture de surface peut se définir comme la capacité des capteurs à surveiller une zone. Les capteurs doivent non seulement couvrir une surface et acquérir des données mais aussi communiquer entre eux pour fusionner, comparer des informations, voir réaliser un multihop jusqu'au Sink. Pour pouvoir exploiter les données collectées, il faut que le graphe soit fortement connexe, c'est à dire qu'il est impératif que tous les nœuds puissent trouver un chemin vers la station collectrice afin de lui transmettre les fruits d'une observation. La couverture de surface

fait l'objet de nombreux sujets de recherche [42] et s'appuie aussi sur des travaux existants tirés de la théorie des graphes.

1.4.3 La contrainte énergétique

Comme mentionné ci-dessus, la contrainte énergétique constitue un grand défi dans les réseaux des capteurs. Une grande partie des études dans ce domaine se focalise sur des techniques d'économie d'énergie pour maximiser le temps de vie de ces réseaux. En effet, ces composants limités en énergie ont des batteries restreintes en taille mais aussi en énergie et souvent non renouvelables, ce qui constitue un handicap considérable lors de l'utilisation de ces composants.

Nous constatons sur la Fig. 1.8 que malgré le développement rapide des processeurs, des mémoires, des technologies sans fil et des capacités de stockage, la capacité énergétique des batteries évoluent à contrario lentement. C'est pour cette raison qu'il existe de multiples techniques pour utiliser les batteries le plus longtemps possible telles que la mise en veille du capteur ou encore l'agrégation de données pour éviter l'envoi de données redondantes.

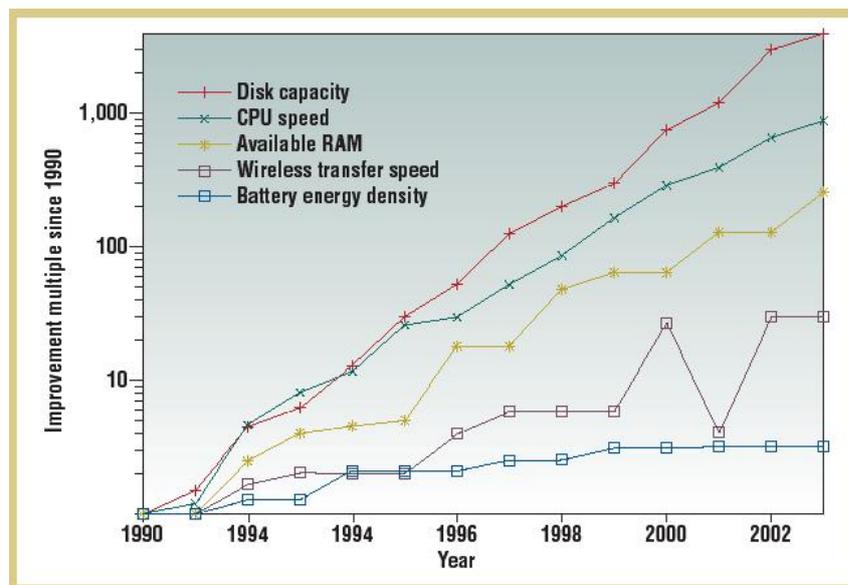


FIG. 1.8 – Evolution des technologies informatique et électronique entre 1990 - 2003 [43]

Après avoir énoncé la contrainte sévère qui est celle de l'énergie, il faut noter que les protocoles de routage utilisés pendant la collecte de données doivent être particuliers. En effet, contrairement aux réseaux *ad hoc* qui sont moins denses et où les nœuds ont leur propre objectif, dans les réseaux de capteurs, les nœuds collaborent pour un objectif commun.

La gestion des flots de données diffère en ce sens que les réseaux de capteurs utilisent une technique *Many to one* tandis que dans les réseaux *ad hoc* c'est du *any to any*. La différenciation au niveau des flots de données et la contrainte d'énergie encore plus sévère chez les capteurs a donné naissance à de nouveaux protocoles de routage pour réseaux de capteurs, dont quelques principaux [44, 45, 46, 47, 48, 49, 50], qui sont définis chacun selon un des trois modes de collecte d'information [51] existant :

- à intervalle régulier : les capteurs envoient régulièrement les résultats des observations au Sink,
- sur demande : la station de base envoie une requête sur le réseau et attend une réponse, c'est le cas le plus fréquent d'application de l'agrégation de données,
- dirigé par évènement : à la moindre détection dans la zone d'observation, le capteur envoie un message à la station de base.

Après avoir défini les RdCs, nous allons nous intéresser aux applications dans lesquelles ils sont impliqués.

1.5 Les applications utilisant les réseaux de capteurs

Dans cette partie nous allons décrire un échantillon d'applications susceptibles d'utiliser des réseaux de capteurs et le cadre dans lequel cette thèse va se placer.

1.5.1 Les applications militaires

Il faut reconnaître que les applications militaires ont souvent été au centre de nombreux développements technologiques. En effet, pour de nombreuses technologies comme ARPANET devenue INTERNET, les réseaux *ad hoc* et aujourd'hui les réseaux de capteurs, le domaine militaire a été le point de départ. Les réseaux de capteurs ont été initiés en 1993 dans le cadre du projet WINS [52] de la DARPA. Dans le domaine militaire, cette technologie peut impulser de nouvelles stratégies de communication [53] ou encore servir à détecter des dispositifs nucléaires et les dépister, à surveiller les activités d'ennemies, ou à analyser un endroit stratégique et difficile d'accès avant un déploiement de troupes.

1.5.2 Les applications médicales

Le domaine médical constitue un intérêt de plus en plus grandissant pour les nouvelles technologies [54]. On utilise de plus en plus de matériels hautement technologiques en médecine. L'apport de ces techniques permet non seulement de garder le patient à son domicile mais aussi de lui éviter le traumatisme de l'hospitalisation. Il faut ajouter qu'à l'aide de capteurs, les comportements anormaux des personnes dépendantes tels que des chutes, des chocs ou des cris peuvent être détectés, ce qui facilitera les interventions immédiates. Dans le même sens, le projet STAR (Système Télé-Assistance Réparti) de l'équipe LIMOS de Clermont-Ferrand en collaboration avec le service de cardiologie du CHU de la même ville propose une plateforme de télésurveillance novatrice permettant de suivre en continu et à distance les personnes ayant des troubles du rythme cardiaque [55].

1.5.3 Les applications liées à la sécurité des infrastructures

Après l'effondrement du pont de Minneapolis dans le Minnesota (USA) en Août 2007, de nombreux budgets ont été déployés pour la recherche sur la surveillance de ponts à l'aide de capteurs. En effet, après cet effondrement une grande polémique s'est développée autour de sa

cause. De ce fait, à l'aide de capteurs, il sera possible de prévoir, voire anticiper la détérioration des ponts dans un univers où le paysage urbain connaît déjà une utilisation avancée des "systèmes de transport intelligents". Par exemple, le projet de transport intelligent Guidestar TMS à Minneapolis a permis une réduction du nombre d'accidents. Les chercheurs s'intéressent également à la surveillance des bâtiments à l'aide de capteurs [56].

1.5.4 La domotique

Nous imaginons très bien les maisons du futur où une véritable interaction avec des capteurs embarqués permettra de contrôler localement ou à distance des appareils domestiques [57]. Ces *smart home*, c'est à dire ces maisons intelligentes, vont faciliter les activités domestiques quotidiennes telles que l'automatisation de l'activation/l'extinction de la lumière qui existe déjà dans certains garages ou encore la mise en marche automatique de la télévision, la climatisation ou le chauffage lorsqu'il y a une présence dans le salon.

1.5.5 Les applications environnementales

Au cœur des débats, l'environnement constitue un axe important pour le devenir des Hommes. L'observer, le comprendre, le connaître a toujours été un but fondamental pour les Hommes et pour la Science. De ce fait, utiliser le progrès technique pour conduire ces différentes actions constitue un atout sans précédent. C'est pour cette raison que les RdcS constituent un nouvel outil pour les applications environnementales.

La mise en place d'un réseaux de capteurs permettra non seulement d'observer notre environnement mais aussi d'anticiper sur d'éventuelles catastrophes naturelles, ce qui constitue une évolution dans un monde qui se voit alerter par la situation de son environnement. La taille des capteurs et leurs autonomie leur donnent la capacité d'être déployés en grand nombre et dans tout environnement, comme ceux qui sont considérés comme sensible et hostiles pour l'homme [58].

D'autre part, cette technologie dans le cadre d'une étude environnementale fait appel à plusieurs compétences, ce qui permettra de réunir plusieurs unités de recherches provenant de domaines différents.

Au cours de cette thèse, nous allons porter un intérêt particulier aux réseaux de capteurs déployés pour des applications environnementales. Il existe différents projets de RdcS déployés pour des applications environnementales [59, 60, 61, 62], dont ceux cités dans cette partie et sur lesquels nous allons mettre l'accent dans le prochain chapitre.

1.6 Conclusion

Ce chapitre a défini les réseaux de capteurs, les outils d'exploitation qui y associés, les technologies de communication utilisées dans ces réseaux émergents et les différentes applications les mettant en œuvre. Parmi ces applications, nous allons porter un intérêt particulier sur les RdcS déployés pour des applications environnementales.

Le prochain chapitre présente un état de l'art sur l'évolution des plateformes utilisées dans les RdcS en présentant quelques projets d'observation environnementales dans lesquels ils ont pu être évalués.

2

Les plateformes et les projets de réseaux de capteurs

Sommaire

2.1	Les plateformes de réseaux de capteurs	36
2.2	Les projets	41
2.2.1	The Wired Integrated Network Sensors (WINS) 1993-1998	41
2.2.2	Smart Dust (Berkeley) 1995-2001	42
2.2.3	Smart-Its	43
2.2.4	The ZebraNet Wildlife Tracker	44
2.2.5	TurtleNet	46
2.2.6	Volcano	47
2.2.7	A macroscope in the redwoods	48
2.2.8	Glacsweb	50
2.2.9	Sensorscope	51
2.2.10	Récapitulatif sur l'évolution des plateformes	52
2.2.11	Le programme HYDRO-SENSOR-FLOWS (2006-2010)	54
2.3	Synthèse sur les projets de réseaux de capteurs	56
2.4	Les architectures hybrides	56
2.4.1	SensEye	57
2.4.2	Surveillance d'un site archéologique	60
2.5	Conclusion	60

2.1 Les plateformes de réseaux de capteurs

Dans cette partie, nous présentons les différentes plateformes pour réseau de capteurs qui existent, leurs avantages, leurs inconvénients, les différents projets dans lesquels elles ont été évaluées avant de proposer notre architecture de réseau de capteurs. Nous allons donc faire un état de l'art sur l'évolution des plateformes de réseau de capteurs.

En effet, le développement rapide dans les domaines de l'informatique, de l'électronique et des technologies sans fil a abouti à l'émergence de petits dispositifs distribués capables de surveiller de manière autonome nos environnements pour nous prévenir des éventuelles catastrophes : les réseaux de capteurs [6]. Ces mêmes progrès scientifiques qui mettent à disposition différentes plateformes de réseau de capteurs (Fig. 2.1), procurent une vision pragmatique à la recherche dans les RdCs [63, 64].

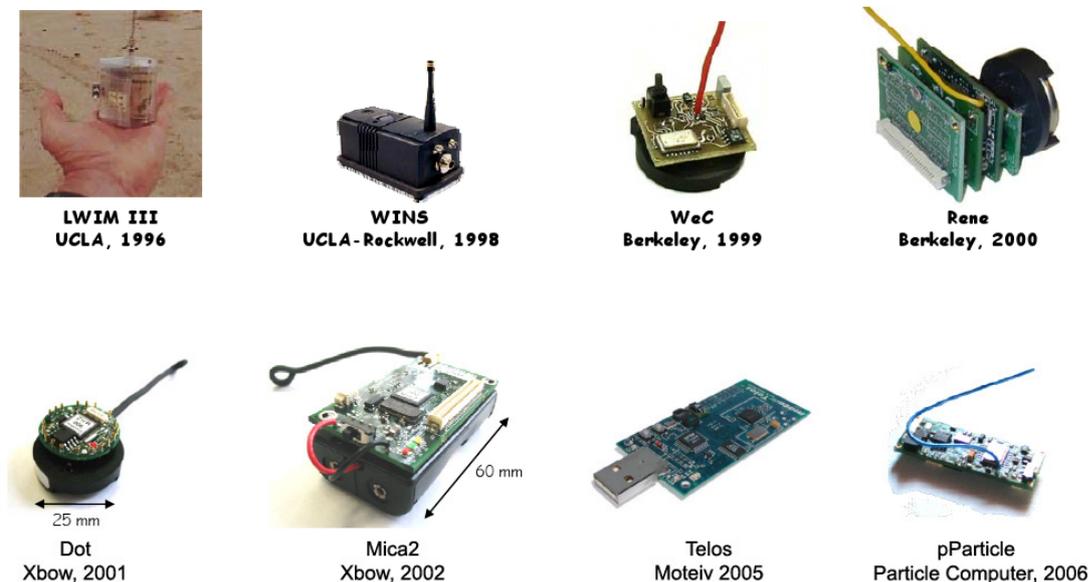


FIG. 2.1 – Les plateformes de réseaux de capteurs [65]

En utilisant ces plateformes, les chercheurs peuvent valider et exploiter des résultats obtenus dans des conditions réelles et ne pas se contenter de résultats fournis par des simulateurs. La Fig. 2.1 illustre bien la disponibilité du matériel, la multiplicité des plateformes et la rapidité de renouvellement de celles-ci où quasiment chaque année une nouvelle plateforme est mise sur le marché. Il existe encore beaucoup d'autres plateformes, telles que le Spec node [66] ou encore les Telos [67] mises sur le marché respectivement en 2003 et 2004, mais pour ne pas surcharger l'image elles n'ont pas été toutes représentées. Cependant, une liste exhaustive est disponible sur le site *The Sensor Network Museum* [68]. Les applications utilisant les réseaux de capteurs étant de plus en plus fréquentes et complexes, le nombre de plateformes aussi se multiplie vu le nombre croissant d'applications visées. Celles-ci diffèrent de par leurs capacités mémoire, leurs unités de traitement, leurs modules de transmission ou encore leurs tailles (Fig. 2.2).

Le tableau de la Fig. 2.2 illustre l'expansion sur le marché des microcontrôleurs ATmega128 [69] et TI MSP430 [67] pour leur consommation énergétique moindre ou encore des modules de transmission CC2420 [70] basés sur le standard IEEE 802.15.4 et conçus pour une consommation moindre.

Plateforme	MCU	RAM	Flash	Stockage	Radio	Taille
Spec node (2003)	AVR RISC 8 bit	3 KB	0 KB	0 KB	RF	2 mm x 2.5 mm
pParticle V2/2x	PIC 18f6720	4 KB	128 KB	512 KB	TR1001	45 x 27 mm
mica (2001)	ATMega128	4 KB	128 KB	512 KB	TR1000	
mica2 (2002)	ATMega128	4 KB	128 KB	512 KB	CC1000	58 x 32 x 7 mm ³ (x 2AA)
mica2Dot (2002)	ATMega128	4 KB	128 KB	512 KB	CC1000	25 x 6 mm (x 2AA)
micaZ (2004)	ATMega128	4 KB	128 KB	512 KB	CC2420	58 x 32 x 7 mm ³ (x 2AA)
Rene2 (2000)	ATMega163	1 KB	16 KB	32 KB	TR1000	
Telos A (2004)	TI MSP430	2 KB	60 KB	512 KB	CC2420	
Telos B (2004)	TI MSP430	10 KB	48 KB	1 MB	CC2420	65 x 31 x 6 mm ³ (x 2AA)
Tmote Sky (2005)	TI MSP430	10 KB	48 KB	1 MB	CC2420	3.2 x 8 x 1.3 cm ³
BTnode3 (2004)	ATMega128	64 KB	128 KB	180 KB	CC1000/ ZV4002 Bluetooth	58.15 x 33 mm (x 2AA)
imote (2003)	ARM7	64 KB	512 KB	0 KB	ZV4002 Bluetooth	
imote2 (2007)	Intel PXA271 XScale	256 KB	32 KB	0 KB	CC2420	36 x 48 x 9 mm ³
Stargate (2003)	Intel PXA255 XScale	64 MB	32 MB		Bluetooth ou IEEE 802.11	9.53 x 6.33 x 1.86 cm ³
La carte Fox (LX 416)	Axis ETRAX 100LX	16 MB	4 MB		Flexible	66 x 72 mm

FIG. 2.2 – Tableau récapitulatif des caractéristiques des plateformes des réseaux de capteurs

Que ce soit pour la famille des nœuds mica en passant au nœud micaZ, des telos ou encore la deuxième version de l'imote, le Chipcon CC2420 est incontournable et donc le standard IEEE 802.15.4 aussi. Cela est dû au fait que ce transceiver soit conforme au standard IEEE 802.15.4 qui est dédié aux RdCs pour sa consommation moindre et aussi sa disponibilité sur la marché car il utilise la bande de fréquence ISM de 2,4GHz qui est libérée dans la plupart des pays du monde.

La course à la miniaturisation fait que les plateformes deviennent de plus en plus petites comme le témoigne la mise sur le marché du nœud Spec node mais dans le même temps les ressources aussi diminuent et face à cela d'autres plateformes de plus haut niveau naissent.

Nous assistons alors à un grand nombre de plateformes de réseaux de capteurs spécifiques à des applications données. Nous remarquons bien sur le tableau de la Fig. 2.2, les capacités extrêmes et diamétralement opposées entre la Spec node et la plateforme Stargate. C'est pour cette raison qu'une classification des plateformes pour réseau de capteurs est proposée où les capteurs sont regroupés selon leurs éventuelles fonctions dans un réseau [71]. Ce regroupement va des plateformes ayant des capacités de calculs et de mémoires réduites à des plateformes plus développées.

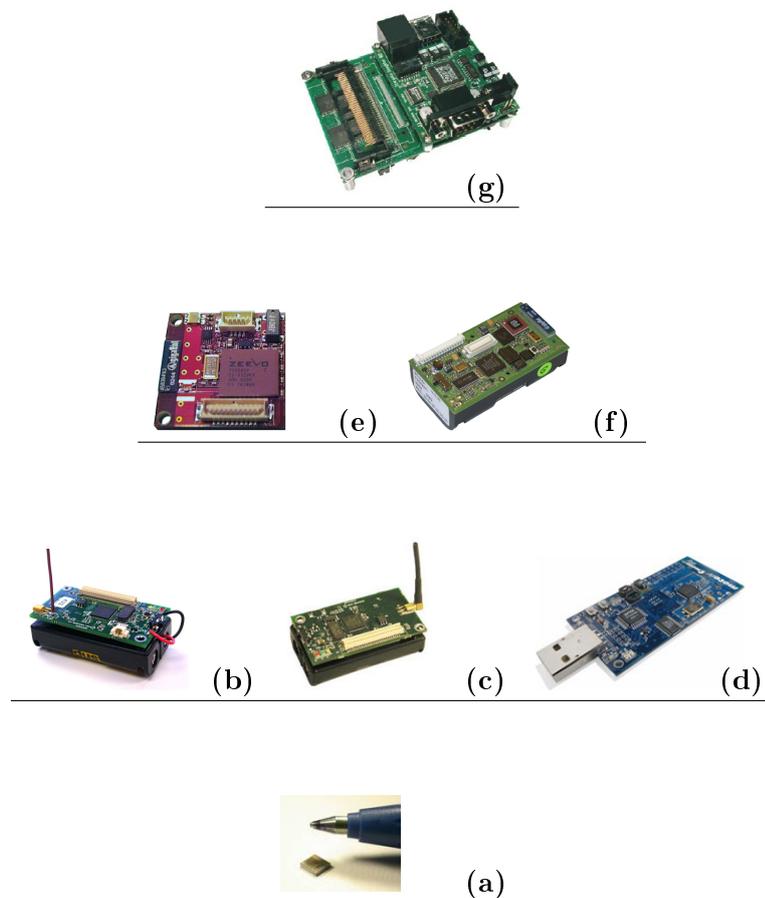


FIG. 2.3 – Classification des plateformes de réseaux de capteurs

Il faut noter que l'énergie consommée est proportionnelle au nombre de portes logiques (CISC²⁰ > RISC²¹, ou encore une architecture 32 bits > 16 bits), à la vitesse (plus il est lent et moins il y a de communication de portes et donc la consommation est réduite) et à la durée de fonctionnement pour exécuter une tâche donnée. De là, nous comprenons aisément que la gestion d'énergie, qui est un point critique dans les réseaux de capteurs, a entraîné le développement de capteurs de très petites puissances limitées par rapport à la multitude d'applications de plus en plus complexes d'où la naissance d'algorithmes adaptés à ces contraintes tels que les algorithmes de compression qui seront présentés dans le Chapitre 4 mais aussi de plateformes de puissance plus élevée pour réaliser des tâches complexes comme celle utilisée dans le cadre de ce travail : la carte Fox (Fig. 2.2).

A partir de là, une hiérarchie au niveau des plateformes de RdCs se dégage. L'architecture (Fig. 2.3) décrit quatre niveaux : des capteurs peu puissants en remontant vers des capteurs de haut niveau. Le premier niveau repose sur le type *Spec node* (a) qui est une puce conçue à

²⁰CISC est un acronyme de Complex Instruction Set Computer c'est-à-dire une architecture matérielle de microprocesseurs à jeu d'instruction étendu

²¹RISC est un acronyme de Reduced Instruction-Set Computer c'est-à-dire une architecture matérielle de microprocesseurs à jeu d'instruction réduit

l'Université de Californie, à Berkeley, pour des opérations de coût ultra-faible et de faible puissance. Ce petit bijou technologique mesure approximativement 2mm x 2.5mm. Il se compose d'un microcontrôleur 8 bit Atmel AVR RISC, une mémoire (3K) et une communication radio basée sur la modulation par déplacement de fréquence (FSK²²). Le niveau qui suit se compose des plateformes Crossbow [10] qui sont les plus populaires et permettent des captures génériques. On peut citer : les motes Mica2 (b), MicaZ (c) et Telos (d).

Les motes Mica2, utilisant des piles de types AA pour fonctionner, sont équipées de plusieurs capteurs : lumière, température, pression barométrique, accéléromètre, et disposent d'un cœur de processeur Atmel AVR. Celles-ci diffèrent des motes micaZ au niveau du module de transmission qui est respectivement pour la mica2 et MicaZ un Chipcon CC1000 et un Chipcon CC2420. La principale différence entre ces deux transceivers est le fait que le Chipcon CC2420 est conforme au standard IEEE 802.15.4/ZigBee tandis que le Chipcon CC1000 utilise les bandes de fréquence 315/433/868 et 915 MHz dont certaines sont interdites dans certains pays du monde tels que le Japon (qui interdit l'utilisation entre 433-915 MHz) ou encore l'Europe (qui interdit la bande de 915 MHz). A cela s'ajoute la différence de portée qui est plus longue par exemple à 433 MHz mais aussi le débit qui est respectivement pour le Chipcon CC1000 et le Chipcon CC2420 de 38,4 Kbps et de 250 Kbps.

La famille des Telos, c'est à dire des nœuds de capteurs avec USB, diffère principalement de celles qui précèdent au niveau du microcontrôleur. Les nœuds de type telos tournent sur des microcontrôleurs fabriqués par Texas Instruments appelé TI MSP430 [72], conçus pour des applications embarquées à basse consommation. C'est un dispositif qui consomme très peu et compatible avec la norme IEEE 802.15.4 pour la communication sans fil. Comme la mote MicaZ son transceiver est de type Chipcon CC2420.

Il y a ensuite les capteurs dotés d'architectures 32 bits et à haut débit qui utilisent la technologie sans fil Bluetooth comme moyen de communication. En effet, contrairement aux plateformes du niveau précédent qui utilisent IEEE 802.15.4 pour échanger leurs données, les plateformes définies ici utilisent Bluetooth qui a un débit plus élevé et donc peuvent transmettre des informations plus volumineuses ou encore le même type information mais plus rapidement. Les principaux nœuds de capteurs dans cette section sont l'imote (e) développé par Intel Research et BNode (f) élaborée à l'ETH de Zurich [73]. Nous pouvons ajouter dans cette section la carte Fox qui s'articule autour d'un processeur 32 bit Axis ETRAX (architecture CRIS²³) cadencé à 100 MHz.

Cependant, les deux plateformes BNode et imote dont les caractéristiques figurent sur le tableau de la Fig. 2.2 ont connu différentes révisions qui sont : l'imote2 et la Bnode3. Les principaux changements pour l'imote sont : le passage d'un ARM7 à un processeur Xscale d'Intel et d'un module de transmission Bluetooth, Zeevo TC2001 Bluetooth, à un chipcon CC2420.

La plateforme BNode par contre qui est une plateforme flexible utilisable dans le domaine des réseaux *ad hoc* en général et dans celui des réseaux de capteurs en particulier, a deux modules de transmission, Bluetooth Zeevo ZV4002 et Chipcon CC1000 (bande ISM 433-915 MHz) et sa revision 3 n'est rien d'autre que la mote Mica2 avec plus de SRAM (256K) et un module radio Bluetooth additionnel (Bluetooth Zeevo ZV4002). La complexité de l'architecture Blue-

²²FSK est un acronyme de Frequency-Shift Keying

²³CRIS est un acronyme de Code Reduced Instruction Set

tooth limite le réseau formé par les BTnodes à 4 Piconets à 7 esclaves, ce qui constitue un réseau vraiment petit de 32 nœuds maximum par rapport à la définition même d'un réseau de capteurs. Cette plateforme est principalement utilisée dans deux projets majeurs : NCCR MICS [74] et Smart-Its [75].

Le dernier niveau est constitué de capteurs à haut débit de type Stargate (g) capable de capturer des informations, les agréger puis les transmettre et ainsi donc de former une passerelle avec d'autres réseaux tel que internet. Il est conçu par Intel et produit par Crossbow. La plateforme Stargate est basée sur un processeur 32 bit d'Intel, PXA-255 XScale à 400 MHz, qui dispose de 32 Mo de mémoire flash, 64 Moctets de mémoire SDRAM, et de connecteurs Crossbow pour les motes mica2 ou micaZ ainsi qu'une entrée PCMCIA pour avoir une interface Bluetooth ou IEEE 802.11.

Cette classification montre bien que la course à la miniaturisation a entraîné des limites au niveau du champ d'application des capteurs, ce qui fait que des plateformes de plus haut niveau avec des processeurs puissants et des technologies à haut débit naissent. On assiste alors à une hiérarchisation des plateformes.

Face à cela, une nouvelle génération de plateformes qui combinent des unités de calculs apparait tel que : la Sun SPOT (Fig. 2.4) [76] qui est une plateforme émergente développée par Sun Microsystems. Celle-ci est disponible depuis avril 2007 et fournit une hiérarchie au niveau des processeurs. De dimensions 41 x 23 x 70 mm, elle est équipée d'un processeur 32 bit ARM9 à 180 MHz avec 512Kb de mémoire RAM, 4Mb de mémoire Flash et d'un microcontrôleur additionnel Atmega48 qui contrôle le RTC (Real-Time Clock) et est chargé de réveiller le processeur principal lorsqu'il est en mode veille. Le transceiver utilisé pour communiquer est le Chipcon CC2420. Les principaux éléments de la Sun Spot sont :

- capteur de température et capteur de luminosité,
- 8 LEDs à trois couleurs chacune et Accéléromètre trois axes 2G/6G,
- 6 entrées analogiques et 2 interrupteurs à retour,
- 5 broches d'entrées/sorties et 4 broches de sortie pour courant élevés.



FIG. 2.4 – Sun Spot

Dans la suite de ce manuel, nous allons présenter les différents projets s'articulant autour de la thématique des réseaux de capteurs. Cette présentation va non seulement illustrer l'engouement pour les chercheurs qui se tournent vers cette nouvelle thématique, mais également fournir une vue d'ensemble sur l'évolution des plateformes utilisées dans ces projets dont celles citées plus haut ainsi que les différents axes de recherche abordés.

2.2 Les projets

Dans cette partie, nous allons présenter quelques projets célèbres de réseau de capteurs. Cette description des projets va illustrer certains dispositifs présentés dans la section précédente au cours de déploiements réels, les applications dans lesquelles ils ont été utilisés, les contraintes de déploiements, l'autonomie dans des conditions réelles et surtout les stratégies utilisées pour accroître cette autonomie, c'est à dire les différents verrous de recherche.

L'objectif de notre travail étant le déploiement pour des applications environnementales, nous allons présenter des applications qui s'inscrivent dans ce cadre. Cependant, cette présentation va aussi nécessiter l'introduction d'autres projets et peut être divisées en trois catégories :

- La première catégorie [52, 77, 78] présente des projets qui sont les premiers dans le domaine et qui ont défini les concepts qui vont accompagner les réseaux de capteurs. C'est à dire qu'à ce niveau nous donnons les définitions qui étaient à la base de ces réseaux et les plateformes qui étaient sensées les accompagner. Cette première catégorie est importante car elle définit les concepts qui accompagnent les RdCs et aussi permet d'identifier les évolutions des dispositifs jusqu'à aujourd'hui.
- la deuxième catégorie présente deux projets originaux (Zebranet [79] et TurtleNet [80]) avec des contraintes sévères mettant en scène des animaux qui interagissent avec leurs environnements respectifs. Ces projets non seulement introduisent notre attachement pour les applications environnementales mais également mettent en valeur deux verrous de recherche qui vont nous intéresser dans cette thèse : la compression de données avec Zebranet et le stockage de données avec Turtlenet.
- La troisième catégorie [58, 59, 60, 61] est le cas précis d'applications environnementales qui nous intéressent, proches du programme Hydro-Sensor-Flows [62] dans lequel est impliqué le département Temps-Fréquence de nos collègues électroniciens qui participent à cette thèse.

Cette observation va non seulement nous familiariser avec les différentes plateformes utilisées dans des conditions réelles, leurs limites, leurs contraintes mais aussi les différents thèmes de recherche abordés dont ceux qui vont nous intéresser.

2.2.1 The Wired Integrated Network Sensors (WINS) 1993-1998

Ce projet de l'Université de Californie [52], UCLA (University of California, Los Angeles), constitue l'un des premiers projets de déploiement d'une plateforme de réseaux de capteurs sans fil. Ce projet a été sponsorisé par l'agence pour les projets de recherche avancée de défense des Etats unis, DARPA. La première génération sur le terrain était en 1996 et la nouvelle génération en 1998. Le but de ce projet était de mettre en place un réseau de capteurs avec une interface internet où la communication se faisait par multi-sauts.

En 1996, les plateformes nommées *Low power Wireless Integrated Microsensors* (LWIM) (Fig .2.5) ont été produites par l'UCLA et le Rockwell Science Center et ont démontré la possibilité d'intégrer plusieurs capteurs avec des interfaces électroniques capables de communiquer dans un même dispositif. Les LWIMs étaient capables de communiquer avec un débit jusqu'à 100 kbps dans un rayon de 10 mètres avec une puissance de 1 mW.

En 1998, la même équipe a développé la deuxième génération, *The Wireless Integrated Network Sensor (WINS)*. Cette nouvelle plateforme se composait d'un processeur 32 bit Intel StrongARM SA1100, 1Mb de SRAM et 4Mb de mémoire flash, un module de communication radio supportant un débit de 100 kbps avec une puissance ajustable de 1 à 100 mW, une batterie et des capteurs sismiques (géophone) et capables de faire du multimedia. La consommation du processeur était de 200 mW en mode actif et 0.8 mW en mode veille.



FIG. 2.5 – LWIM III

Tandis que ce projet offrait une plateforme relativement puissante avec des possibilités de communication, une autre philosophie est apparue à l'université de Berkeley en Californie s'intéressant à des plateformes de très petites tailles consommant moins d'énergie.

2.2.2 Smart Dust (Berkeley) 1995-2001

Ce projet qui a donné naissance au concept de poussière intelligente avait pour but de mettre en place un système de la taille d'un grain de sable, c'est-à-dire un millimètre cube capable d'être déployé en réseau de milliers de nœuds [77]. Ces poussières électroniques communicantes sont formées de quatre principales unités compactées : un module capture, un module de traitement, un module de communication et un module gestion d'énergie.

La Fig. 2.6 montre les principaux composants d'un smart dust : le module d'alimentation composé de batterie et de cellules photovoltaïques, les capteurs et le module de transmission garantissant une liaison par laser optique. La technique du laser a été privilégiée par rapport à la RF qui consomme plus d'énergie.

En 1999, ce concept de l'université de Berkeley a abouti au premier nœud de la série des motes : *WeC* (Fig. 2.7). La plateforme WeC se composait d'un microcontrôleur 8 bit d'Atmel à 4 MHz, 512b de RAM et 8 Kb de mémoire flash, qui consommait 15 mW en mode actif et 45 μ W en mode veille, un module radio TR1000 offrant un débit jusqu'à 10 Kbps avec 36 mW de puissance en émission et 9 mW en réception et des capteurs de température et luminosité.

La philosophie générale cachée derrière ce concept est le fait que plus un nœud est petit, plus nous pouvons en déployer, ce qui permettra d'observer de manière efficace un environnement même si au fil du temps certains d'entre eux deviennent défectueux car il en restera toujours pour observer. Aussi, la taille facilite le déploiement car plus des nœuds sont petits plus ils sont faciles à déployer. C'est pour cette raison que des motes avec la même philosophie, *Rene* (1999) et *Dot* (2000), ont suivi avec des microcontrôleurs plus performants (Fig. 2.8).

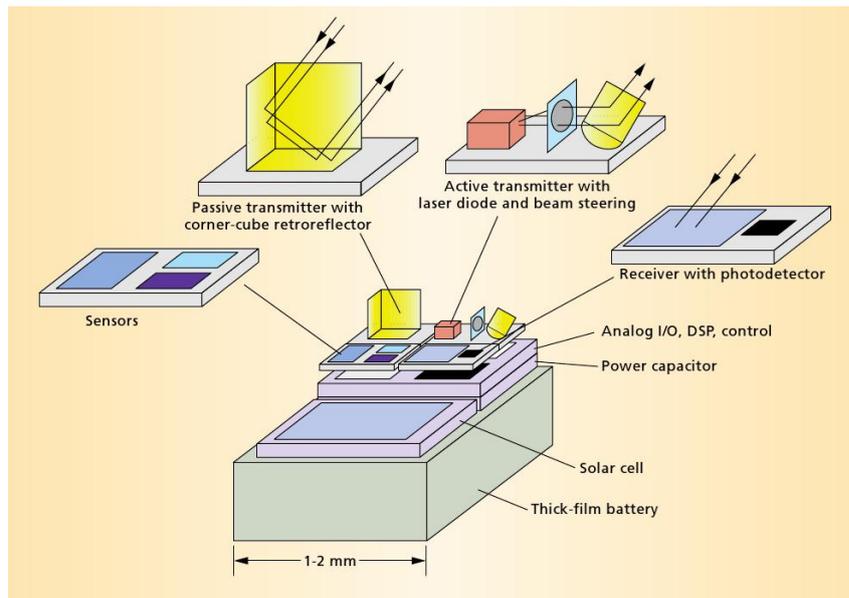


FIG. 2.6 – Les principaux composants d'un smart dust [77]

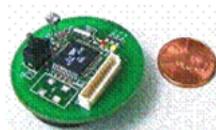
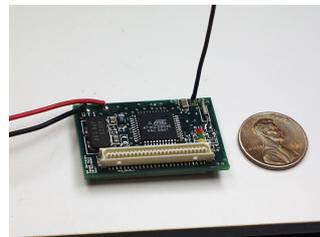


FIG. 2.7 – WeC



(a)



(b)

FIG. 2.8 – Les motes Rene (a) et Dot (b)

2.2.3 Smart-Its

Le projet Smart-Its [78] qui est une collaboration entre plusieurs universités Européennes : l'université de Lancaster, l'ETH Zurich, l'université de Karlsruhe, Interactive Institute et VTT s'inscrit dans le cadre d'une initiative européenne et est financé en partie par la Commission de l'Union européenne, et l'office fédéral de l'éducation et de la Science Suisse. Ce projet s'intéresse à une vision ambitieuse de calcul embarqué dans le monde. Dans cette vision, les divers objets de tous les jours sont équipés de médias, capables d'entrer en relation numérique dynamique. Dans ce projet, on considère de petits systèmes embarqués qui peuvent être attachés à des objets du quotidien afin d'augmenter leur détection, leur perception, leur possibilité de calcul et communication. Les plateformes BTnodes ont été utilisées dans ce projet.

Les présentations qui précèdent vont au delà même de la notion de simple projet car elles définissent des concepts qui vont accompagner les recherches dans les réseaux de capteurs. En effet, on constate dans ces projets le souhait de définir et valider des concepts nouveaux qui vont donner des directions à cette thématique.

2.2.4 The ZebraNet Wildlife Tracker

ZebraNet [79] est un projet inter-disciplinaire qui a permis de conjuguer les efforts de différents laboratoires tels que le Département d'informatique (Dept. of Electrical Engineering) et le Département de Biologie (Dept. of Ecology and Evolutionary Biology) de l'Université de Princeton. L'objectif de ce projet est de déployer des capteurs sur des Zèbres de la réserve de Sweetwaters à MPALA en plein centre du Kenya.

D'un point de vue purement informatique, ce projet vise à explorer différents nœuds de recherche tels que les protocoles sans fil, les techniques de stockage non volatile, le positionnement des nœuds etc. Les objectifs sont donc de développer, d'évaluer, de mettre en oeuvre, des systèmes de test qui intègrent le calcul, la communication sans fil et le stockage non volatile sur des nœuds de capteurs équipés de récepteurs GPS.

Vu sous l'angle des biologistes, l'objectif est d'utiliser le système de façon à améliorer les nouvelles études sur les migrations d'animaux et les interactions entre les espèces. En tant que problème de recherche des systèmes informatiques, ZebraNet est convaincant parce que les besoins des chercheurs en biologie sont suffisamment sévères pour exiger de véritables percées dans la conception de systèmes informatiques, de protocoles sans fil ou encore dans la gestion énergétique des systèmes conçus. Parmi les contraintes de départ fixées par les biologistes il y avait :

- Le fait de prélever les positions GPS toutes les trois minutes. De prime abord, la durée de trois minutes a été choisie par les biologistes parce que des travaux de recherche ont démontré que cette durée est assez longue pour enregistrer des statistiques comportementales indépendantes et encore assez fréquente pour avoir suffisamment de points.
- Une année de fonctionnement sans aucune intervention humaine directe.
- Le système doit être capable de fonctionner sur une grande surface d'une centaines à milliers de km².
- Pas de station de base fixe car des stations fixes petites ou grandes font l'objet de vandalisme.
- Tandis que la latence n'est pas un point critique, le fait de collecter l'intégralité des fichiers logs est très important.
- Pour le collier, une taille comprise entre 3-5 lbs est recommandée (Soit entre 1,36-2,27 Kg). Pour des animaux plus petits la limite serait encore plus petite.

Le premier déploiement a eu lieu en Janvier 2004 sur 7 nœuds (6 femelles et 1 mâle) dans la réserve Sweetwaters au Kenya sur une surface de 100 km² [81]. Durant ce déploiement, les positions sont stockées toutes les 8 minutes. Par contre, toutes les deux heures chaque nœud cherche un autre nœud dans les alentours capable de faire suivre les données collectées jusqu'à la station de base.

Il faut noter que dans ce projet, les nœuds du réseau, "*les zèbres*", sont mobiles et la station

de base aussi. On considère que les chercheurs collectent les informations lorsqu'ils passent à proximité par avion ou par voiture. C'est l'une des premières études dans le domaine prenant en compte simultanément la mobilité des nœuds et du Sink. La Fig. 2.9 décrit le matériel utilisé dans cette expérience :

- Un Microcontrôleur TI MSP430F149 16 bit avec 2KB RAM et 60 KB de mémoire Flash. Une des principales raisons de ce choix est le fait qu'il supporte plusieurs fréquences d'horloge. On utilise l'horloge à 8 MHz pendant que le GPS, le transceiver ou le module flash fonctionne sinon on bascule sur l'horloge à 32 KHz pour économiser de l'énergie.
- Un module radio MaxStream 9XStream (902-928 MHz). Il a été choisi pour pouvoir communiquer sur de longues distances. Ce module utilise la bande de fréquence de 900 MHz et offre la possibilité de communiquer dans un rayon jusqu'à 8 Km dans des conditions extérieures idéales.
- Une mémoire flash : ATMEL 4MB AT45DB041B.
- Un module GPS μ -blox GPS-MS1E .
- Une série de 14 panneaux solaires produisant chacun 7 mA à 5V qui chargent une batterie lithium-ion.

Il faut noter que ces découvertes peuvent être utilisées pour d'autres problématiques de recherche pas seulement orientées sur l'étude des espèces, car les contraintes des Rdc ont été prises en compte. Les résultats de l'évaluation énergétique montrent que la transmission est l'activité qui consomme le plus : la transmission (780 mW), le GPS (568 mW), la réception (312 mW) et enfin le microcontrôleur a une consommation négligeable par rapport aux autres composants entre 9,6 mW à 32KHz et 19,3 mW à 8MHz. De là, une autre problématique est apparue : qui est l'organisation du stockage d'information à l'aide de la compression de données pour éviter l'envoi de données systématique et permettre un stockage plus efficace sur un espace réduit. Ce qui a donné naissance à un célèbre algorithme de compression dans les réseaux de capteurs : S-LZW [82] qui est une adaptation pour les systèmes embarqués du fameux algorithme LZW [83]. Cet algorithme est actuellement l'algorithme de référence sur la compression de données dans les réseaux de capteurs. Cependant, une partie de cette thèse sera consacrée à la définition de nouvelles techniques de compression capables de fournir des meilleurs taux de compression que le célèbre algorithme S-LZW avec une consommation énergétique moindre.

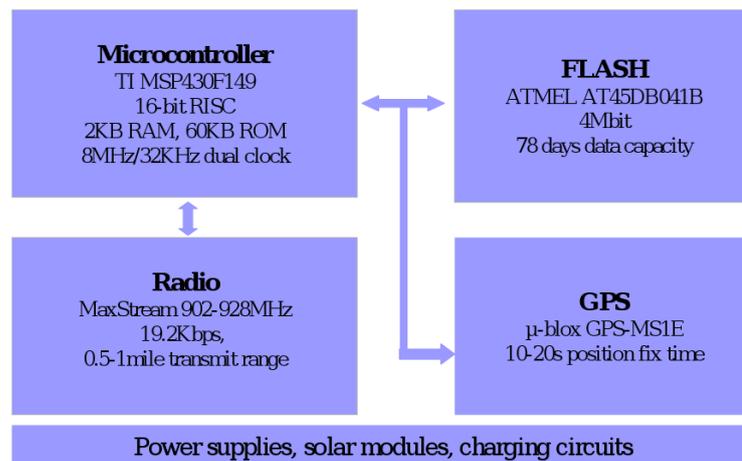


FIG. 2.9 – Description de la plateforme de Zebranet

Aussi, ce projet a démontré les limites rencontrées pendant le déploiement. En effet, un jour après la mise en place des colliers (Fig. 2.10), les zèbres semblaient être affectés et les biologistes ont constaté plusieurs mouvements de la tête. Mais à la fin de la première semaine, il y a eu finalement peu de différence comportementale entre les zèbres avec et sans colliers.

Le plus gros problème technique rencontré pendant ce déploiement était la communication radio, spécifiée à 8 km et testée dans le New Jersey à environ 1,6 km, qui finalement a affiché une courte distance de communication variant entre les différents colliers. Cette variation était de 100 mètres à 1,2 km pour les colliers testés au Kenya. Ces résultats illustrent bien la complexité de cette thématique où les résultats théoriques dans la pratique ne sont pas validés du premier coup et le manque de stabilité des techniques de communication sans fil.

Le but de ce projet ambitieux est de permettre aux chercheurs en biologie d'étudier la migration à longue distance des espèces, leurs interactions et le comportement nocturne.



FIG. 2.10 – Zebranet

2.2.5 TurtleNet

Ce projet comme son nom l'indique a pour but d'étudier le comportement d'une espèce menacée, les tortues, à l'aide d'un réseau de capteurs [80]. Cette expérience, effectuée le long du canal de *Deerfield* à l'Ouest du Massachusetts, a réuni au sein d'un même projet deux équipes de chercheurs de domaines différents : une équipe d'informaticiens qui a testé les communications réseau et l'autre formée de biologistes qui a étudié le comportement d'une espèce bien précise de tortues : les chélydres. En effet, l'aménagement du territoire fait que leur espace se réduit.

L'idée principale de ce projet est de créer un réseau d'appareils autonomes (fonctionnant sur des batteries qui s'auto-rechargent) qui se déplacent constamment, qui enregistrent et stockent l'information, puis transmettent des données d'un périphérique à un autre jusqu'à une station de base. C'est dans cette optique que les chercheurs de l'Université du Massachusetts ont créé le projet TurtleNet : le premier réseau formé uniquement de tortues (Fig. 2.11) qui a été financé par des subventions de la *National Science Foundation*. D'un point de vue logiciel, le stockage d'informations a été réalisé à l'aide d'un environnement appelé Capsule [84]. Ce système a été développé sous TinyOS à l'Université de Massachusetts (Umass) Amherst et supporte le stockage sur mémoire Flash (technologie NAND ou NOR).

Environ 15 tortues vivant près du canal, équipées de dispositifs permettant de les localiser et de connaître leur température corporelle, ont participé au réseau TurtleNet. Le matériel utilisé est la mote Mica2Dot de Crossbow équipée du récepteur GPS SiRF Star III pris d'un Deluo ca-

pable de capter un signal GPS dans des conditions extrêmes, de batterie Lithium rechargeable : Ultralife UBC581730 250 mAh et de panneaux solaires de type : 4,2V powerfilm flexible.

Dans un déploiement plus récent des améliorations ont été faites telles que la mote Mica2Dot qui a été remplacée par une TinyNode [85] pour accroître les performances radio ou encore l'utilisation de panneaux solaires plus efficace (6V Solarbotics SCC3733). La Tinynode, de taille 30x40 mm, s'articule autour d'un microcontrôleur TI MSP430 (MSP430F2417) cadencé à 16 MHz et est doté d'un transceiver qui opère sur la bande ISM de 868/915 MHz et consomme peu : le Semtech SX1211 (3 mA en mode réception et 25 mA en mode transmission à +10dBm).



FIG. 2.11 – Photo d'une tortue faisant partie du projet Turtlenet

Dans le cadre de notre travail (Chapitre 5), nous présenterons aussi un système de stockage d'information sur carte SD développé sous TinyOS-2.x. Notre environnement d'enregistrement d'information est basé sur une adaptation du système de fichier FAT sous TinyOS, ce qu'il lui procure la possibilité d'être multi-plateforme. Cet avantage permet de lire directement le support de stockage, en l'occurrence ici une carte SD, sur un ordinateur sans aucun traitement logiciel supplémentaire, ce qui simplifie le travail de l'utilisateur final qui retire simplement son support de stockage du capteur et l'analyse directement sur son ordinateur.

2.2.6 Volcano

Ce projet interdisciplinaire étudie l'utilisation des réseaux de capteurs sans fil pour la surveillance des éruptions des volcans qui sont actifs et dangereux [58]. L'interdisciplinarité de ce projet a uni les travaux des informaticiens de l'Université d'Harvard et des volcanologues de différentes universités qui sont : l'Université de North Carolina, l'Université du New Hampshire et l'institut géophysique de l'Equateur. En effet, le potentiel des réseaux de capteurs sans fil peut largement améliorer la compréhension de l'activité volcanique.

Trois réseaux de capteurs sans fil ont été déployés sur les volcans actifs en respectant l'architecture de la Fig. 2.12. Le premier déploiement, qui se composait d'une petite gamme de capteurs, a eu lieu en Juillet 2004 au volcan Tungurahua, en Équateur, et a servi de preuve du concept en collectant de manière continue des données infrasonores.

Le second déploiement en Août 2005 au volcan Reventador, en Equateur, se composait de 16 nœuds (Tmote Sky, Fig. 2.13) déployés sur 3 km qui ont mesuré des signaux sismiques et acoustiques pendant 3 semaines. De là, 230 événements volcaniques ont pu être collectés.

Le troisième déploiement, à Tungurahua en août 2007, était axé sur l'utilité des données collectées par le système développé afin d'améliorer la fidélité des données.

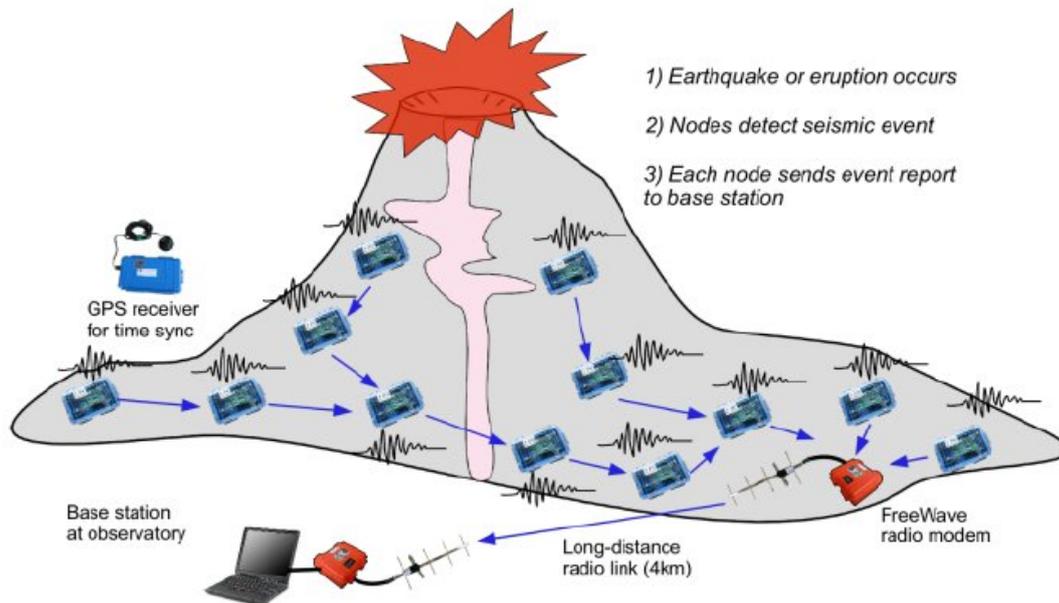


FIG. 2.12 – Architecture de surveillance des volcans

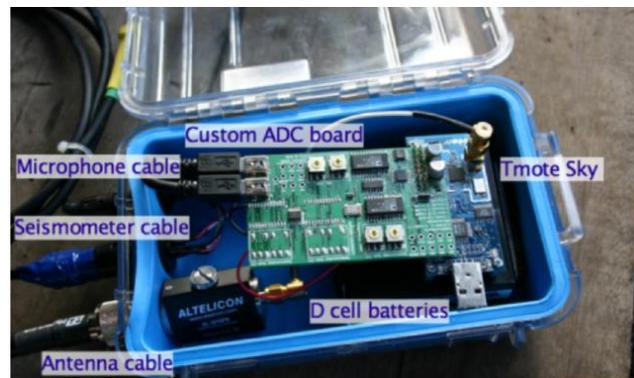


FIG. 2.13 – Capteur sismique et acoustique sans fil

2.2.7 A microscope in the redwoods

Ce projet [59], réunissant les informaticiens et les biologistes de l'Université de Berkeley en Californie, offre une observation spatiale et temporelle dense. Ce projet répond à une certaine méthodologie de questionnement. En effet, lorsque les biologistes et les informaticiens se sont réunis, la question posée aux biologistes par les informaticiens était la suivante : qu'est ce que vous voulez observer qui simplement n'a pas été mesuré jusqu'à aujourd'hui. La classification des réponses a donc abouti à l'étude de l'Ecophysiologie d'un arbre de type Redwood à Sonoma en Californie.

Ces chercheurs, grâce à un réseau de capteurs collectant des informations toutes les 5 minutes pendant 44 jours, ont pu observer un arbre Redwoods de 70 mètres.

La Fig. 2.14 décrit le positionnement des capteurs, placés approximativement tous les deux mètres sur le Redwood, collectant la température, l'humidité et le rayonnement solaire photosynthétiquement actif et le matériel utilisé dont les principaux éléments sont :

- une mote Mica2Dot de Crossbow,
- deux photodiodes de type Hamamatsu S1087 pour mesurer le rayonnement solaire photosynthétiquement actif incident et réfléchi,
- un capteur TAOS TSL2550 pour mesurer le rayonnement solaire total,
- un module Intersema MS5534A pour mesurer la pression barométrique.

La passerelle entre les motes et l'extérieur est une plateforme Stargate équipée d'un modem GPRS.

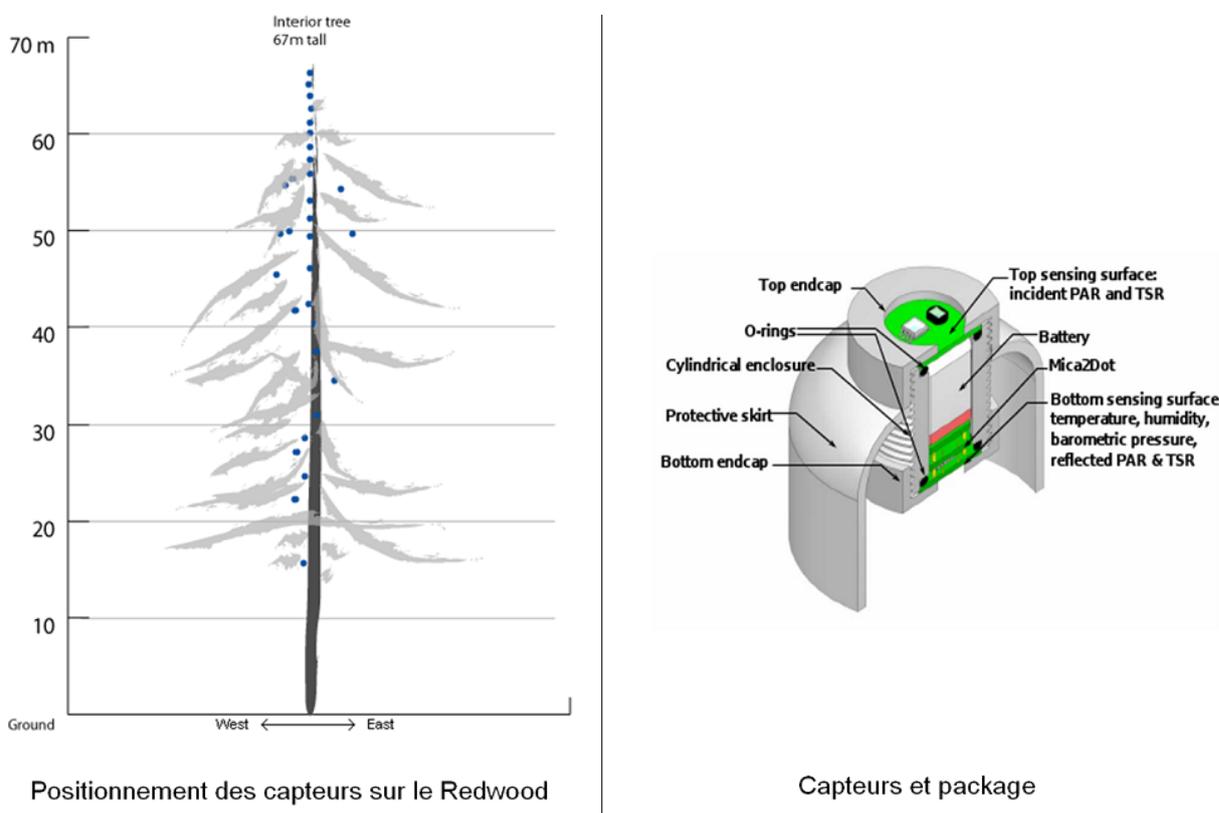


FIG. 2.14 – Description de l'architecture générale de Macroscopic

Ce système s'appuie sur une architecture logicielle appelée TASK [86], développée par l'université de Berkeley et le laboratoire Intel de Berkeley, permettant d'interagir à l'aide d'une framework sur les données collectées par les capteurs et rappatrier au niveau de la passerelle, ce qui facilite le suivi du réseau. Une architecture similaire avec un noeud concentrateur qui reçoit des informations provenant de noeuds qui collectent de simples données scalaires sera présentée dans le prochain chapitre.

2.2.8 Glacsweb

L'objectif de ce projet [60], qui est une collaboration entre les laboratoires d'informatique et de géographie de l'Université de Southampton, a pour but de développer une technologie permettant d'observer les glaciers en utilisant les réseaux de capteurs. Ce projet a ainsi contribué à la recherche fondamentale en glaciologie et dans le monde des réseaux de capteurs. Ce projet a abouti au premier déploiement en 2003 à Briksdalsbreen en Norvège (Fig. 2.15).



FIG. 2.15 – Photo du glacier observé à Briksdalsbreen en Norvège

Ce déploiement comprenait 9 capteurs installés dans le glacier à une profondeur allant de 50 à 80 m (Fig. 2.16). Pour observer le comportement durant la fonte du glaciers, les capteurs décrit sur la Fig. 2.17 ont collecté l'inclinaison, la température et la pression.

Ce système est capable de fonctionner pendant plusieurs mois. Les résultats des données captées montrent que pendant que la pression et la température d'un nœud diminuent tous les 7 jours, le degré d'inclinaison reste le même.

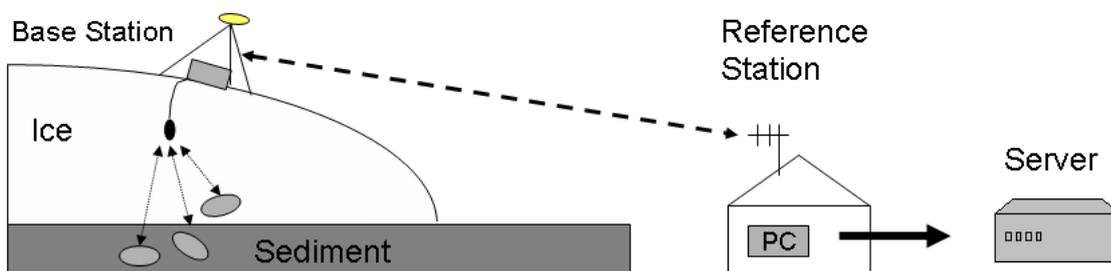


FIG. 2.16 – Architecture générale du réseau

L'architecture matérielle utilisée dans ce projet (Fig. 2.17) est inhabituelle par rapport aux projets précédents car elle s'articule autour d'un microcontrôleur PIC et nous notons aussi l'utilisation d'une liaison radio à 433 MHz.

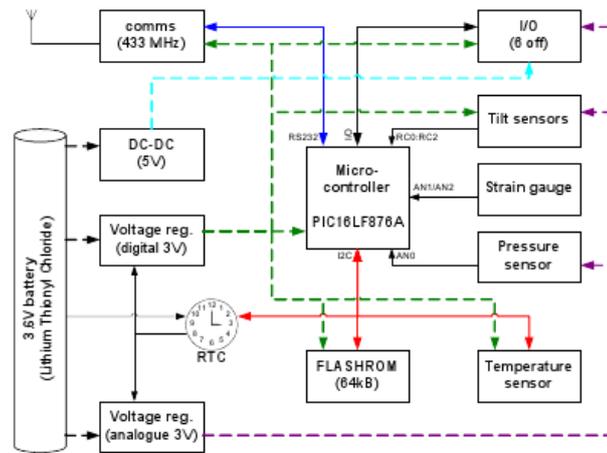


FIG. 2.17 – Architecture des capteurs (version 2) déployés dans le glacier

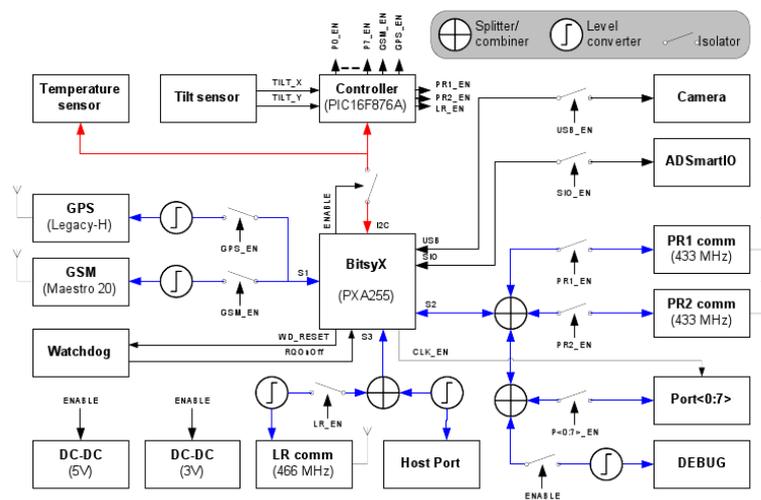


FIG. 2.18 – Architecture de la station de base

2.2.9 Sensorscope

Le projet multidisciplinaire Sensorscope [61] développé à l'Ecole Polytechnique Fédérale de Lausanne (EPFL), a pour but de définir une plateforme de réseaux de capteurs facile à déployer et s'intéressant à la surveillance environnementale. Plus précisément, un déploiement s'étalant sur deux mois a été effectué dans un glacier rocheux en Suisse pour étudier les coulées de boue qui sont dangereuses. Ce système de surveillance permet d'avoir en temps réel via internet les données collectées par les capteurs et a nécessité la définition d'une pile de communication complète sous TinyOS [87]. Le matériel utilisé dans ce réseau est la mote TinyNode qui a une grande portée radio, 200 m en extérieur, avec une faible consommation de l'ordre de 25 mA en mode transmission (0 dB). Les données collectées par les capteurs sur chacune des stations (Fig. 2.19) sont : la température de l'air et du sol, l'humidité de l'air, le taux d'absorption et l'humidité du sol, la vitesse et la direction du vent, radiation solaire et précipitations.

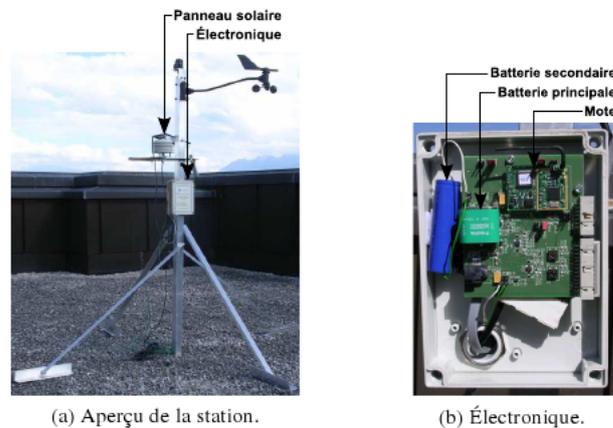


FIG. 2.19 – Une station de collecte SensorScope

Pour une meilleure autonomie du système, le module énergétique est formé de trois principaux composants, un panneau solaire et deux batteries. Les caractéristiques du panneau solaire de taille 162x140 mm sont une durée de vie de 20 ans pour une puissance de 1W en éclairage direct. La batterie principale de type NiMH utilise le panneau solaire pour se recharger et a une capacité de 150 mAh. La deuxième batterie de type Li-Ion est une batterie de secours ayant une capacité de 2200 mAh qui prend le relais de la première lorsque celle-ci atteint un certain seuil et est rechargée par la batterie principale lorsque celle-ci est à nouveau pleine. Dans ce projet, 16 stations ont été déployées sur une surface de 500 x 500 m² pendant deux mois.

2.2.10 Récapitulatif sur l'évolution des plateformes

L'analyse des projets qui précèdent montre que les plateformes Mica2Dot et TinyNode sont très utilisées. Le point commun apparent entre ces plateformes est le module de communication qui utilise les bandes de fréquences de 433,868/916 MHz ou 310 MHz pour la Mica2Dot et 868/915MHz pour la TinyNode. La caractéristique principale de ces bandes est la possibilité de transmettre des informations sur une longue distance. Nous constatons donc que malgré le fait que la technologie IEEE 802.15.4/ ZigBee (bande ISM de 2,4 GHz) soit très répandue, beaucoup de déploiements réels opèrent sur des bandes qui permettent une transmission à longue distance.

Par contre, il faut noter qu'au niveau logiciel, la complexité algorithmique qu'offre IEEE 802.15.4/ZigBee, tel que le routage, est bien meilleure que celle des radiomodems qui ne font que du point à point pour communiquer comme la liaison RS232.

Cependant, la transmission reste encore une préoccupation et un point d'interrogation lors des déploiements de RdCs qui se traduit, par exemple, par les difficultés rencontrées par le projet Zebranet. En effet, les technologies sans fil sont contraintes par plusieurs facteurs tels que les conditions environnementales, ce qui fait que leur utilisation n'est pas toujours aisée. Cette présentation a aussi montré la multiplicité des axes de recherches proposés par les différents projets tels que le stockage d'information (Turtlnet et Zebranet) en utilisant respectivement des mémoires flash ou encore la compression de données, la transmission d'information (Zebranet), l'autonomie des dispositifs indispensables pour la plupart des projets, etc.

Le tableau 2.20 résume la présentation des projets de réseaux de capteurs en regroupant les plateformes utilisées, les modules de transmission, le nombre de nœuds déployés, l'autonomie des réseaux et les différents verrous de recherche abordés. Nous remarquons également que la majorité des applications déjà déployées ne se limitent qu'à une dizaine de nœuds contrairement à la définition même d'un réseau de capteurs.

Projet	Unité de calcul	Radio	Nombre de nœuds	Autonomie	Axe de recherche
Zebranet	TI MSP430F149	MaxStream 9XStream (902-928 MHz)	7	1 an (envisagé)	Protocole de routage, techniques de stockage et positionnement.
Turtlenet	Atmel ATmega128L (Mica2Dot)	transceiver multi-channel (433, 868/916, ou 310 MHz)	15		Protocole de routage, techniques de stockage et positionnement.
Volcano	TI MSP430F1611 (Tmote Sky)	Chipcon CC2420	16	3 semaines	Protocole de routage, développement logiciel et positionnement.
Macroscope	Atmel ATmega128L (Mica2Dot)	transceiver multi-channel (433, 868/916, ou 310 MHz)	Tous les 2 m (arbre 70 m)	44 jours	Observation spatiale et temporelle dense.
Glacsweb	PIC16LF876A	liaison radio à 433 MHz	9	Plusieurs mois (envisagé)	Développement logiciel et communication.
Sensorscope	TI MSP430F1611 (TinyNode)	Semtech XE1205 (868 MHz)	16	2 mois	Protocole de communication.

FIG. 2.20 – Tableau récapitulatif des projets de réseau de capteurs

Il existe encore beaucoup d'autres projets et plateformes, avec une évolution rapide (Fig. 2.21), qui ont marqué l'histoire des réseaux de capteurs, nous pouvons citer les références suivantes :

- PicoRadio(MIT) [88],
- μ Amps (MIT) [89],
- Wisenet (CSEM) [90],
- Great Duck Island (Berkeley) [91],
- COMMON-Sense Net (EPFL) [92],
- Permasense [93].

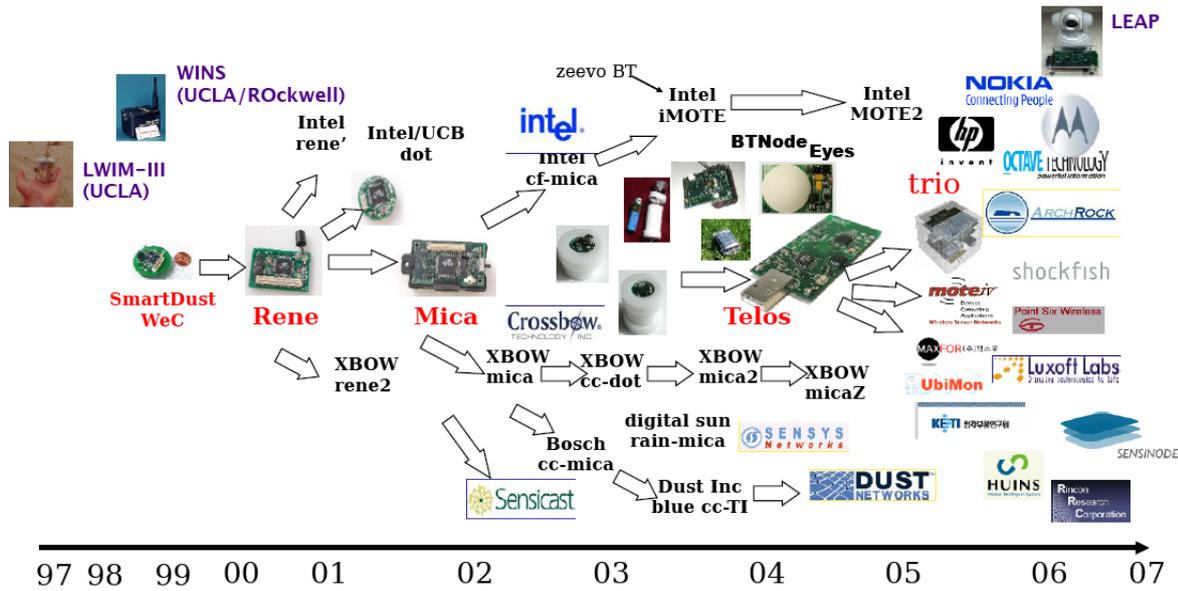


FIG. 2.21 – Evolution du phénomène des réseaux de capteurs [94]

Dans la prochaine section, nous allons porter une attention particulière à un programme où différents laboratoires de l'université de Franche-Comté participent pour le déploiement d'un réseaux de capteurs dans le bassin du glacier Loven Est (Spitsberg, 79°N) [62].

2.2.11 Le programme HYDRO-SENSOR-FLOWS (2006-2010)

Dans cette partie, nous présentons un projet récent qui est la mise en place d'un réseau de capteurs par des équipes Allemande, Espagnole, Russe et différents laboratoires français qui sont :

- Thema et FEMTO-ST de l'université de Franche-comté,
- I.D.E.S. de l'université Paris-Sud (Orsay),
- SET de l'université de Pau et des Pays de l'Adour.

Le but de ce projet est d'observer et quantifier les flux liquides et solides du glacier Loven Est (Spitsberg, 79°N), étendu sur une surface de 10 km², à l'aide d'un réseau de capteurs dense. Ce réseau se compose de :

- 2 stations météorologiques automatiques,
- 9 anémomètres et pluviomètres,
- 30 capteurs thermiques,
- 3 sondes multiparamètres pour le suivi physico-chimique de la qualité des eaux d'écoulement,
- 3 préleveurs automatiques d'eau,
- 10 stations photo automatiques,
- mesures mobiles par Ground Penetrating Radar (GPR) : cartographie du réseau sous-glaciaire, structure du permafrost.

Dans la description qui suit nous allons nous focaliser sur le travail d'une équipe qui se place aussi dans le cadre de cette thèse, le département Temps-Fréquence du laboratoire FEMTO-ST, et plus précisément sur les stations photo automatiques spécialement développées pour ce projet. Ces appareils sont répartis de façon à couvrir l'intégralité du bassin et permettront de suivre, depuis le sol :

- la dynamique du manteau neigeux et son évolution
- la dynamique fluvio-glaciaire
- le positionnement de l'isotherme 0°C sur le glacier pour déterminer les zones de fonte
- la dynamique des versants.

Certains processus très rapides comme la débâcle de printemps au front du glacier pourront être enregistrés et visualisés en continu pour la première fois au Spitsberg.

Face aux conditions environnementales sévères, ces stations photo ont été mises dans des boîtes pour résister au froid (Fig 2.22).



FIG. 2.22 – Capteurs déployés au Spitsberg équipés d'appareils photo numérique

La contrainte énergétique et le désir de maximiser l'autonomie des dispositifs ont poussé à recouvrir ces boîtes de panneaux solaires capables de fournir de l'énergie. Les conditions environnementales et les interférences avec un radiotélescope aux alentours font que la communication sans fil n'est pas une solution fiable. A cela s'ajoute le flux de données à transférer. En effet, pour les 8 stations déployées avec 3 prises de photos par jour sachant qu'une photo fait plus de 3 MB, cela ferait 10 MB par appareil chaque jour soit 80 MB de données au total à transférer tous les jours, ce qui est impossible pour une liaison faible consommation en 115200 bauds car il faudrait 2h.

De là, une alternative à la transmission de données sous forme de stockage sur carte SD a été choisie. Cette solution est non seulement plus fiable mais aussi moins gourmande en énergie car la transmission est l'activité la plus coûteuse en énergie pour un capteur, encore plus lorsque le flux de données est aussi conséquent que celui qui précède. C'est pour cette raison que dans le cadre de cette thèse, nous allons mettre l'accent non seulement sur des techniques d'organisation du stockage de l'information telle que le développement de nouvelles techniques de compression

de données mais aussi sur le développement à l'aide de TinyOS d'un nouvel environnement de stockage d'information sur carte SD utilisant le système de fichier FAT16 pour faciliter à l'utilisateur final l'analyse des données collectées quelque soit le système d'exploitation qu'il utilise.

Les 8 capteurs utilisés dans le programme HYDRO-SENSOR-FLOWS sont équipés de micro-contrôleurs faible consommation TI MSP430 et d'appareils photo numérique de haute résolution. Ce système fonctionne de manière autonome depuis 18 mois pour 3 prises de photo par jour.

2.3 Synthèse sur les projets de réseaux de capteurs

L'état de l'art sur les projets utilisant les réseaux de capteurs montre que la plupart des réseaux déjà déployés sont des projets liés à l'environnement. Pour des applications un peu plus complexes comme la surveillance des glaciers qui nécessite des images de hautes résolution, les plateformes classiques et les outils à faible consommation qui les accompagnent comme les caméras faible consommation sont rapidement dépassés.

Un grand nombre de réseaux, y compris Internet, les réseaux *ad hoc* et aujourd'hui, les réseaux de capteurs, ont commencé par des projets impulsés par l'armée et qui s'étendent par la suite jusqu'aux infrastructures supportant le multimédia. Cependant, de prime abord, le multimedia et les réseaux de capteurs sont antagonistes du fait des ressources limitées des plateformes de réseaux de capteurs. En effet, les capteurs doivent avoir une faible consommation d'énergie tandis que le multimedia est gourmand en énergie. La course à la gestion énergétique a poussé au développement de plateformes ayant des ressources limitées qui s'oppose au succès des réseaux de capteurs pour des applications de plus en plus complexes.

Une solution pour accommoder faible consommation et performances importantes requises par le multimedia est d'utiliser un système hybride. Ce système hybride peut être vu de deux façons :

- soit en cumulant des unités de calculs sur une même plateforme [76, 95, 96], c'est à dire en définissant une hiérarchie de processeurs sur une même plateforme,
- soit en définissant un modèle à plusieurs niveaux où les plateformes sont indépendantes.

Dans ce dernier modèle qui fera l'objet du prochain chapitre, les éléments du niveau le plus bas collectent de simples données scalaires et les dispositifs de plus haut niveau réalisent des tâches plus complexes.

2.4 Les architectures hybrides

Il existe plusieurs plateformes de réseaux de capteurs. Chacune d'entre elle a des caractéristiques qui lui sont propres comme son architecture c'est-à-dire : son unité de calcul, sa mémoire RAM ou encore la technologie sans fil qu'elle utilise. La complexité des nouvelles applications et le grand nombre de plateformes existantes a entraîné la conception de plateformes hybrides de réseaux de capteurs. Une activité complexe qui utilise cette architecture est celle destinée au transport de flux multimedia. En effet, la course à la gestion énergétique a provoqué une limitation au niveau des applications réalisables par les plateformes les plus populaires comme celles de Crossbow.

Celles-ci par exemple utilisent pour faire du multimedia des modules CMOS intégrés sur des caméras Cyclops [97]. Cependant, les modules CMOS qui consomment peu, offrent malheureusement des images de qualité limitée [98, 99]. De là, une architecture de référence a été définie (Fig. 2.23). Dans cette architecture, on distingue différents schémas dont celui qui nous intéresse : une architecture à plusieurs niveaux.

Un exemple de plateforme de haut niveau capable de réaliser du multimedia à l'aide d'une webcam est la plateforme Panoptes [100]. Panoptes a été conçue pour observer l'environnement ou encore pour la surveillance à l'aide de vidéo. Cette plateforme s'articule autour d'un système Linux embarqué équipé d'un processeur Intel StrongARM, de 64 Mb de mémoire, d'une interface 802.11 pour la communication radio et d'une webcam de type Logitech pour la capture de flux multimedia. Cette plateforme offre une possibilité d'adaptation des priorités au niveau des flux vidéos. Pour aller plus loin dans la description des plateformes multimedia pour réseaux de capteurs, nous allons présenter un projet célèbre dans le domaine basé sur une architecture hybride et hiérarchique appelé SensEye.

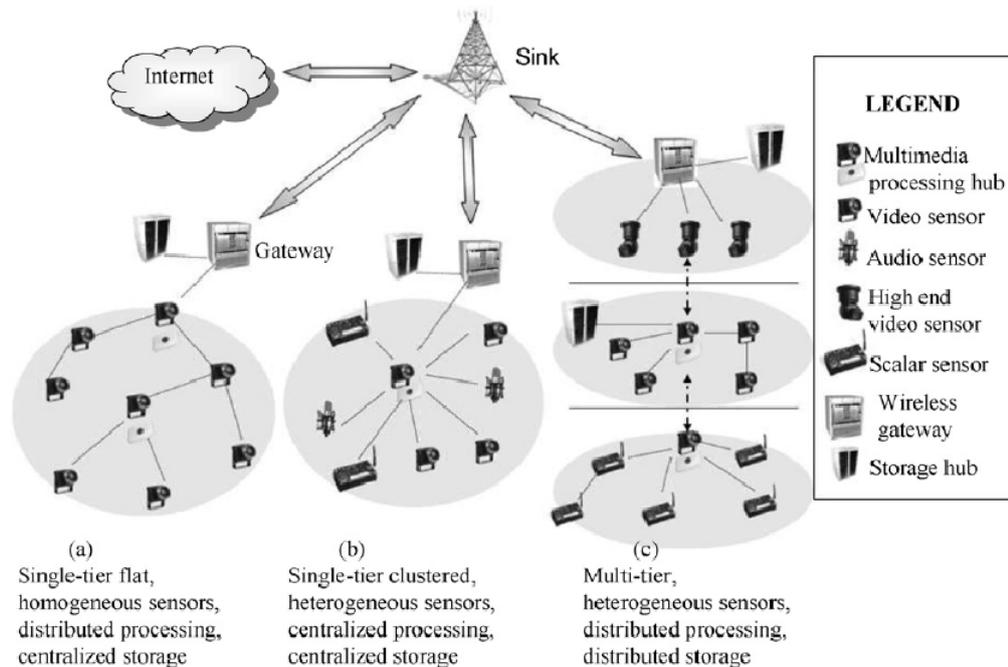


FIG. 2.23 – Architecture de référence pour le multimedia dans les RdC [101]

2.4.1 SensEye

Dans ce projet [102] basé sur une architecture à plusieurs niveaux, l'hétérogénéité ne se situe pas seulement au niveau des nœuds utilisés dans le réseau mais aussi au niveau des caméras et des technologies sans fil. Le projet Senseye définit une architecture hiérarchique à trois niveaux pour les applications de surveillance. Ce type d'architecture offre plus de fonctionnalités qu'un système homogène du fait des caractéristiques diverses de chacune des plateformes à chacun des niveaux. Il faut noter qu'à chaque niveau les fonctionnalités des nœuds sont différentes et c'est l'interaction entre les différents niveaux qui fait la richesse de cette architecture. La Fig. 2.24

illustre l'architecture hiérarchique et hétérogène de SensEye.

Nous constatons que le niveau le plus bas se compose de motes (Mica2) collectant de simple données scalaires tandis qu'au second niveau les motes sont équipés de caméras de qualité moyenne (Cyclops ou CMUcam) quand au troisième niveau se trouve des capteurs Stargates avec des webcams de type Logitech Quickcam Pro de bonne résolution. Un niveau additionnel composé de PC embarqué (mini-ITX) et de caméras de haute résolution et directionnelles de type Pan-tilt-zoom (PTZ) est aussi défini (Fig. 2.25). Deux technologies radios sont utilisées au niveau de la plateforme Stargate, 802.11 et une technologie à bas débit pour communiquer avec les niveaux plus bas qui utilisent exclusivement une technologie à bas débit.

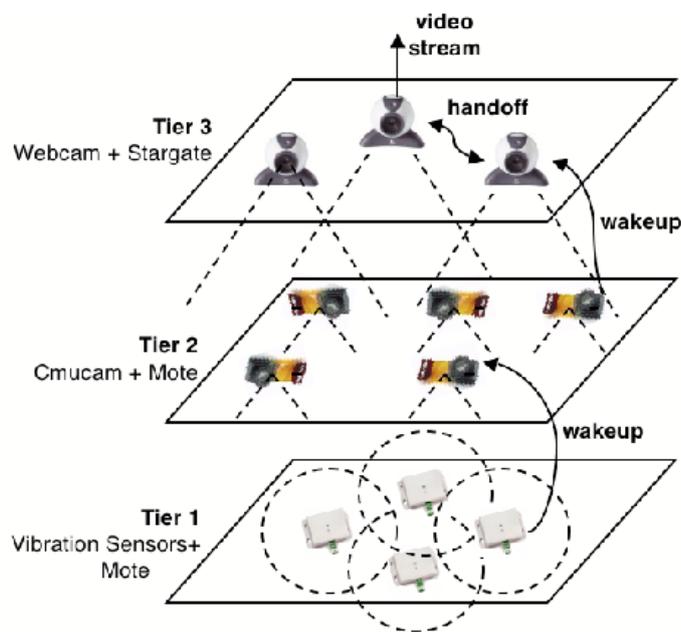


FIG. 2.24 – Architecture hiérarchique de Senseeye [103]

Les deux principales applications qui ont motivé ce travail sont : la surveillance d'espèces rares dans des forêts lointaines ou encore la surveillance des tâches de gestion des catastrophes. De là, les principales caractéristiques de cet environnement sont :

1. La détection d'objets : cette étape consiste à détecter un nouvel objet dans l'espace de surveillance.
2. La reconnaissance (ou l'identification) d'objets : lorsqu'un nouvel objet est détecté, il doit être classifié pour déterminer son type afin de savoir s'il y a un intérêt de le suivre. On considère qu'une base de données des images d'intérêt est disponible par défaut ce qui permet de faire une comparaison avec un nouvel objet dans la zone de surveillance.
3. Le suivi d'objets : en supposant que l'objet détecté est intéressant pour l'application, il peut être suivi quand il se déplace dans l'environnement.

Ce suivi entraîne plusieurs tâches :

- le calcul de l'emplacement actuel de l'objet et de sa trajectoire,
- le transfert de la responsabilité du suivi de l'objet lorsqu'il est hors de la portée visuelle à un autre dispositif de capture qui est dans le même rayon que lui,

- la prise de vidéo ou encore d’une séquence d’images de l’objet puis stockage sur un dispositif ou transmission à une station de base.

Une illustration de l’implication de chacun des niveaux par rapport aux caractéristiques précédentes est défini sur la Fig. 2.25.

L’objectif est de mettre au point une architecture matérielle et logicielle pour accomplir ces tâches de manière à optimiser la consommation d’énergie, sans sacrifier la performance des indicateurs tels que la latence et la fiabilité. En effet, pour des raisons d’économie énergétique, les nœuds des niveaux les plus hauts sont mis en veille et lorsqu’un capteur de niveau plus bas détecte un objet, il ne réveille que le nœud qui est dans le même rayon que l’objet d’intérêt (Fig. 2.24).

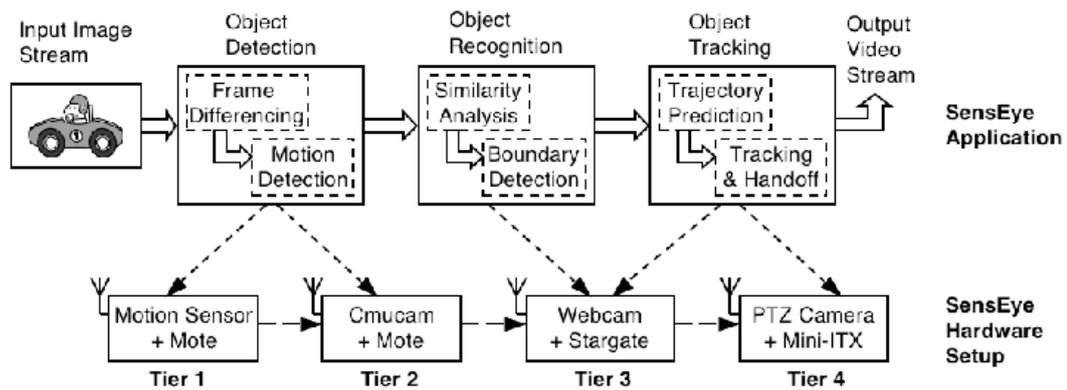


FIG. 2.25 – Scénario de détection, reconnaissance et suivi d’objets [103]

Par contre, en considérant le cas critique où plusieurs objets d’intérêt sont détectés simultanément, comme par exemple un cas exceptionnel qui est la détection d’un troupeau d’animaux rares, tout de suite tous les nœuds du réseau sont impliqués et la consommation du réseau explose.

Aussi, on constate que les deux premiers niveaux de la Fig. 2.25 ont pour objectifs la détection d’obstacle. Les capteurs de vibrations détectent d’éventuels vibrations lorsque qu’un objet s’approche de la zone de surveillance pendant que les capteurs équipés de caméras à faible consommation et qualité moyenne telles que les Cyclops ou les CMUcams font de la détection de mouvement pour voir s’il y a un changement dans l’environnement. L’utilisation de deux niveaux pour la détection d’obstacle permet d’être plus précis et peut être plus rapide dans la détection d’obstacle mais par contre faire de la détection de mouvement à l’aide de caméras de type Cyclops ou CMUcams peut engendrer un coût énergétique supplémentaire.

Pour terminer, il faut noter que la mise en veille des nœuds de niveau supérieur qui consomment le plus et réalisent des tâches complexes peut entraîner la réaction lente de ces capteurs et donc le fait de ne pas pouvoir suivre l’objet à temps. En effet, il faut 4 secondes au nœud de niveau supérieur pour réaliser toutes ses tâches, avec une consommation de l’ordre de 4,71 Joules, dont 366 ms juste pour le simple réveil. Cette latence est aussi due au système Linux qui a besoin d’un certain temps pour changer d’état.

Par contre, la multiplication de niveaux dans une telle architecture qui spécialise certes les caractéristiques de chaque niveau peut avoir aussi une influence sur les communications dues à

l'encombrement des fréquences résultants ainsi à des interférences et encore à la multiplication d'écoute passive qui est aussi une source de consommation énergétique. C'est pour cette raison que dans la prochaine section nous allons nous intéresser à une architecture qui n'a que deux niveaux.

2.4.2 Surveillance d'un site archéologique

Ce projet de l'université de Parlermo définit une nouvelle architecture qui utilise la technologie des réseaux de capteurs pour surveiller le parc archéologique de *la Vallée des Temples* à Agrigente en Italie [104]. En effet, il est vivement souhaité pour un site archéologique d'être surveillé pendant les heures d'ouverture pour observer les comportements anormaux mais aussi pendant les heures de fermeture pour vérifier la non intrusion de personnes malveillantes. Le système proposé, décrit sur la Fig .2.26, comprend deux composants principaux :

1. *L'architecture réseau* qui décrit les plateformes utilisées par le système pour surveiller le site,
2. *Le système logique* qui analyse les informations collectées à l'aide d'une interface graphique pour que l'administrateur prenne la décision adéquate face à une anomalie.

L'architecture réseau de surveillance est une architecture hybride et hiérarchique composée de nœuds MicaZ et de nœuds Stargate équipés de caméras (Fig. 2.27). Le niveau bas formé de nœuds MicaZ collecte des données scalaires comme la température, l'humidité, le bruit, les vibrations etc...

Le niveau haut composé de nœuds plus performants appelés Stargate peuvent ainsi développer des applications multimedia. Dans ce travail, l'accent est mis sur l'analyse des informations collectées, ce qui fait que la gestion énergétique n'est pas le plus important. Certes, les auteurs font référence à une stratégie pour économiser l'énergie au niveau bas de leur architecture lors de la collecte d'information [105], tandis que le niveau haut est alimenté en permanence, mais ils ne donnent aucun détails sur une éventuelle évaluation expérimentale. En effet, l'évaluation dans [105] est faite à l'aide d'un simulateur.

2.5 Conclusion

Ce chapitre a mis en évidence l'évolution des plateformes de réseau de capteurs et les différents contextes dans lesquels elles ont pu être évaluées. Cette description réalisée à travers plusieurs projets a montré l'intérêt pour les applications environnementales par les chercheurs du domaine. Le programme Hydro-Sensor-Flows se démarque par rapport aux autres car il est en cours de réalisation et fait intervenir le département Temps-Fréquence du laboratoire FEMTO-ST qui s'inscrit aussi dans le cadre de cette thèse.

Le programme Hydro-Sensor-Flows servira de repère de par sa multitude d'axes de recherche telles que la conception de plateformes capables de fonctionner dans des conditions extrêmes, le transport de flux multimedia et les difficultés d'utilisation de la radiofréquence. Ces difficultés ont été à la base de cette thèse en proposant des solutions alternatives comme le stockage de données et de la compression de données.

L'état de l'art qui précède montre que les auteurs ne se contentent pas seulement de définir une architecture logicielle mais aussi une architecture matérielle. C'est pour cette raison que dans le prochain chapitre nous allons présenter également notre architecture matérielle et logicielle.

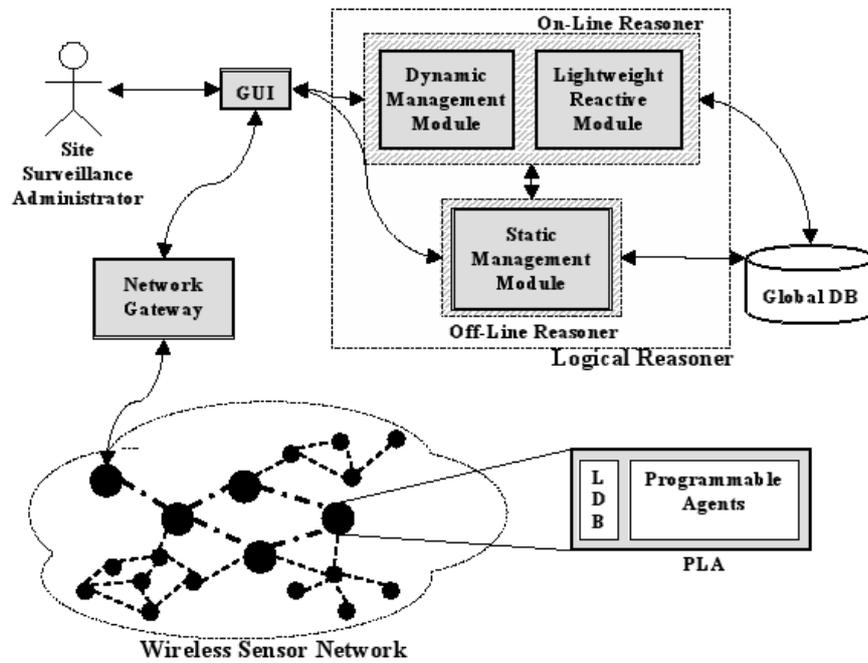


FIG. 2.26 – Architecture générale de surveillance du site archéologique d'Agrigente

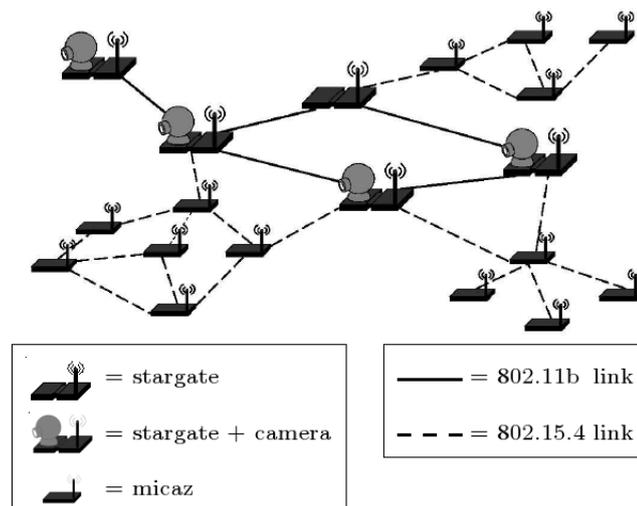


FIG. 2.27 – Architecture hybride et hiérarchique pour la surveillance du site archéologique d'Agrigente

Deuxième partie

Mes contributions

Une architecture hiérarchique et générique de réseau de capteurs

”Tout ce qui est en haut est comme ce qui est en bas et tout ce qui est en bas est comme ce qui est en haut.” Hermès Trimegiste

Sommaire

3.1	Présentation de l’architecture générale	66
3.2	Évaluation des performances du niveau supérieur de notre architecture	67
3.2.1	La carte Fox	67
3.2.2	Le module d’évaluation énergétique	68
3.2.3	Description de l’architecture réseau pour le transfert d’images	69
3.2.4	Résultats expérimentaux	71
3.3	Synthèse sur le niveau supérieur de notre architecture	74
3.4	Présentation du moyen de communication entre les différents niveaux	75
3.4.1	La Tmote Sky	75
3.4.2	Le protocole IEEE 802.15.4	76
3.5	Notre architecture hiérarchique et générique de réseaux de capteurs	78
3.5.1	Implémentation du protocole IEEE 802.15.4 sur notre plateforme	79
3.5.2	Evaluation du compromis énergie/délai du protocole IEEE 802.15.4	83
3.5.3	Introduction de l’ α -pénalité	85
3.6	Conclusion	88

Dans ce chapitre, nous développons notre proposition d'architecture matérielle et logicielle à deux niveaux qui nous permet de saisir des informations scalaires mais également de fournir des informations de qualité supérieure.

3.1 Présentation de l'architecture générale

Comme nous l'avons déjà souligné, l'autonomie des capteurs qui est en général liées aux batteries non remplaçables fait que la gestion d'énergie a toujours été un point critique dans ces réseaux. Ce point critique a entraîné le développement d'une multitude de plateformes équipées de matériels à faible consommation et jusqu'à aujourd'hui seuls les dispositifs à capacités réduites consomment peu. En effet, les besoins et les contraintes des capteurs, font que la grande majorité des plateformes ne sont pas adaptées pour des activités complexes c'est pour cette raison que nous allons mettre l'accent dans ce travail sur les architectures hiérarchiques et hybrides où les plateformes classiques de bas niveau collectent de simples données scalaires et les plateformes de haut niveau sont capables de faire des tâches plus complexes comme du multimédia par exemple.

Différents travaux de recherche se sont déjà focalisés sur les plateformes hiérarchiques et hybrides [102, 104]. Cependant, celles-ci sont spécifiques au multimédia et c'est pour cette raison que dans ce travail nous allons définir une nouvelle architecture matérielle et logicielle qui est générique c'est-à-dire qui n'est pas uniquement destinée aux activités multimédia mais à toutes les activités nécessitant une série de tâches complexes en prenant en compte l'économie d'énergie. La Fig. 3.1 illustre notre architecture générale qui peut s'adapter à différents contextes tels que ceux nécessitant des supports de stockage de masse non volatil.

Dans la partie qui suit nous allons présenter l'architecture générale de notre plateforme. C'est une architecture hiérarchique à deux niveaux où le niveau haut se compose d'un système Linux embarqué sur la carte Fox qui va jouer le rôle de concentrateur d'informations. La carte Fox est l'élément fournissant la plus grande puissance de calcul avec des périphériques rapides. Elle est équipée de deux ports USBs et d'une interface Ethernet capable de faire la liaison avec l'extérieur en utilisant Internet par exemple. Le niveau bas est constitué de nœuds classiques à faible consommation tels ceux fabriqués par Crossbow.

Notre travail se décompose de la manière suivante : dans un premier temps, nous allons évaluer le transfert d'une information riche comme une image qui sollicite des capacités de calculs importantes à l'aide de la technologie sans fil Bluetooth. La seconde partie sera consacrée à l'étude du standard IEEE 802.15.4 qui sera le moyen de communication entre les deux niveaux de notre architecture. En effet, le protocole IEEE 802.15.4 est communément utilisé par les plateformes de RdCs de bas niveau, ce qui montre la flexibilité de la carte Fox par rapport aux différentes technologies. Dans le même temps, nous allons évaluer le compromis énergie/délai associée à cette technologie et donner des conclusions qui vont nous pousser à nous concentrer sur des techniques de stockage d'information telles que la compression de données. Le chapitre 5 détaille une partie de notre travail qui se place dans un contexte différent de ceux qui précèdent, où les conditions environnementales rendent impossible l'utilisation de technologies sans fil. De ce fait, nous allons présenter un espace de notre architecture générale qui est consacré à l'utilisation d'un nouvel environnement de stockage d'information sur support de masse non volatil telle qu'une carte SD.

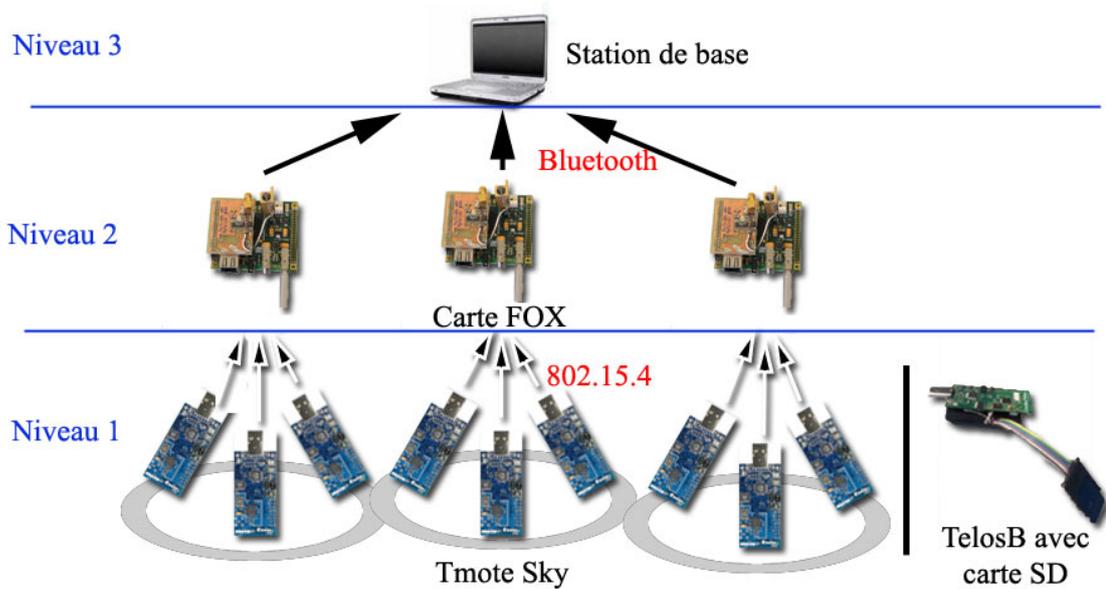


FIG. 3.1 – Notre architecture générale de réseau de capteurs

3.2 Évaluation des performances du niveau supérieur de notre architecture

L'intérêt pour les RdCs utilisant une architecture hiérarchique est de fournir une quantité d'information riche et inaccessible par les capteurs scalaires. En effet, les changements climatique critiques, les applications militaires et la surveillance des comportements ont donné naissance aux RdCs utilisant le multimedia. Cependant, ce nouvel axe de recherche de prime abord difficilement imaginable face aux capteurs peu puissants et à faible consommation a entraîné le développement d'une nouvelle catégorie de capteurs de plus haut niveau comme la carte Fox dont les performances vont être évaluées ici.

3.2.1 La carte Fox

La carte Fox, de dimensions 66 x 72 mm, est un support pour le développement de systèmes embarqués, développée par ACME [16], sur lequel tourne un système d'exploitation GNU/Linux (noyau 2.4 ou 2.6). Ce système fonctionne avec un microprocesseur Axis ETRAX 100LX fournissant 100 MIPS [106] pour une consommation de l'ordre de 1,5 W. Cet environnement, illustré sur la Fig.3.2 supporte de nombreuses applications telles que par exemple un serveur web, des interfaces réseau (ethernet, wifi, bluetooth, ZigBee, GSM etc) avec l'ensemble des développements se faisant sur des outils opensource.

En effet, un atout intéressant de cette plateforme est sa flexibilité, on peut l'adapter à différentes interfaces de communication grâce à ses deux ports USBs sur lesquels il est possible de connecter des dongles Wifi, Bluetooth, ZigBee ou encore une webcam et son port Ethernet permet une connectivité immédiate sur un bus TCP/IP (Internet). A cela, s'ajoute le fait qu'il utilise un système Linux embarqué, ce qui constitue un environnement familier pour les informaticiens. Cet avantage, contrairement à des outils spécialement dédiés aux capteurs, facilite la

portabilité du code existant, réduit considérablement le temps de développement et les risques d'erreurs. La série de la carte avec laquelle nous travaillons est la carte Fox LX 416 avec 4 MB de mémoire flash et 16 MB de mémoire RAM.



FIG. 3.2 – La carte Fox [16]

Les projets [102, 104] utilisent la plateforme Stargate comme plateforme de haut niveau, cependant, celle-ci est en fin de production actuellement. De ce fait, la carte Fox qui est basée sur des outils opensource, reste ouverte et favorable au développement. La prochaine section présente notre module d'observation énergétique ajouté par nos collègues du département Temps-Fréquence du laboratoire FEMTO/ST.

3.2.2 Le module d'évaluation énergétique

La mesure de la puissance consommée nécessite, par une tension d'alimentation régulée supposée fixe, une mesure du courant d'alimentation I . La chute de tension aux bornes d'une résistance de charge $R' = 0,1 \Omega$ fournit cette mesure par amplification différentielle (Fig 3.3). Cette mesure est convertie en flux de données numériques par un convertisseur qualifié d'intelligent puisqu'il embarque un cœur de microcontrôleur 8051. Il faut noter que nous faisons apparaître une première hiérarchie puisque ce coprocesseur ADuC814 qui décharge le processeur principal de la gestion de la mesure de courant peut aussi être configuré pour réveiller la carte Fox sur détection d'un évènement extérieur tel que décrit auparavant par la bibliographie.

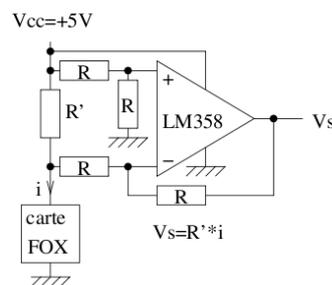


FIG. 3.3 – Schéma de l'analyseur d'énergie

Cette stratégie d'analyse énergétique a été influencée par les perspectives d'évolution de notre plateforme pour des applications où le coprocesseur permettrait non seulement de connaître la consommation du processeur principal mais aussi réagirait en conséquence selon les différents niveaux de consommation pour d'éventuelles mise en veille. De là, un accès direct à la consommation par un coprocesseur ADuC814 a été choisi. Aussi, il faut noter que la conception de plateformes hybrides devient de plus en plus répandue [76, 95, 96].

La résistance de 0.1 ohms insérée sur l'alimentation de la carte Fox permet donc une mesure du courant du circuit en cours de fonctionnement. La chute de tension associée au courant

circulant dans la résistance est mesurée par le convertisseur courant-tension à base d'amplificateur opérationnel alimenté de façon unipolaire, selon le circuit présenté sur la Fig. 3.3. Le microcontrôleur équipé d'un convertisseur analogique numérique 12 bits (ADuC814) se charge de l'échantillonnage des mesures et du transfert au rythme d'environ 5 Hz des données à la carte fox par une liaison asynchrone série (RS232). Un récepteur GPS qui n'est pas utilisé dans cette expérience a aussi été ajouté pour des déploiements futurs (Fig 3.4).

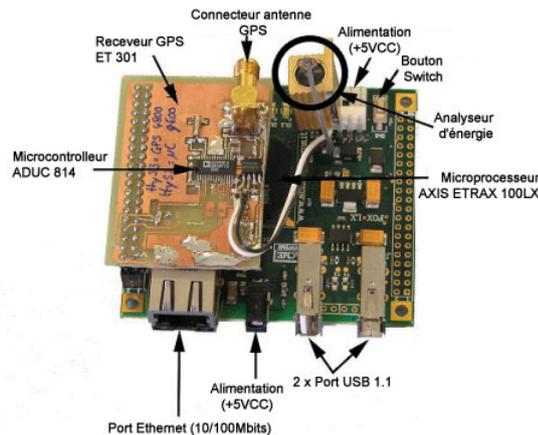


FIG. 3.4 – Notre carte Fox expérimentale

La comparaison des Fig. 3.2 et Fig. 3.4 montre les modifications électroniques apportées par nos collègues électroniciens du département Temps-Fréquence du laboratoire FEMTO/ST.

3.2.3 Description de l'architecture réseau pour le transfert d'images

Le choix d'une technologie sans fil est important car il influe sur la consommation énergétique et l'architecture réseau. Nous allons donc évaluer la consommation énergétique et les délais pendant la transmission de plusieurs images JPEG de différents tailles en utilisant des interfaces Bluetooth de classe I et de classe II sur un réseau point à point. Pour atteindre cet objectif nous avons dû ajouter à notre carte Fox expérimentale différents dispositifs comme le module énergétique qui précède pour évaluer la consommation énergétique, une webcam pour prendre des images et des dongles Bluetooth pour pouvoir transférer les images.

En général, la technologie sans fil utilisée par les nœuds de haut niveau d'une architecture hiérarchique est le standard IEEE 802.11 [102]. Cependant, le principal inconvénient de ce standard pour les RdCs est sa consommation énergétique considérable. C'est pour cette raison que nous allons nous focaliser sur Bluetooth qui fait partie des WPANs et constitue un bon compromis entre les standards IEEE 802.11 et IEEE 802.15.4 respectivement pour la consommation énergétique et le débit.

Au début Bluetooth a été conçu pour remplacer les câbles dans des réseaux de proximité, ce qui fait que le passage à l'échelle n'a pas été une priorité, d'où sa complexité pour une architecture étendue. Cependant, comme décrit dans [107], les chercheurs portent de plus en plus d'attention à la formation des scatternets [21, 23, 22]. D'autres travaux de recherche dans les réseaux de capteurs utilisent la technologie Bluetooth [108]. Une plateforme célèbre utilisée dans différents travaux [108, 73] est la plateforme BTnode équipée d'une interface Bluetooth. Les auteurs dans [109] utilisent aussi les BTnodes et concluent que malgré les contraintes de la

technologie Bluetooth, elle présente des caractéristiques intéressantes comme un bon ratio de consommation énergétique par bits transmis.

Nous allons donc ici analyser les performances de nos capteurs en simulant un réseau simple point à point à l'aide de trois cartes Fox (Fig. 3.5). L'accent va être mis sur le nœud 1 (*grabber*) qui acquiert des images à l'aide d'une webcam, puis les envoie et sur le nœud 2 (*router*) qui sert de relais. Le nœud 3 est la station de base (Sink). L'objectif de cette étude est de router en multisaut des images sur réseau Bluetooth pour pallier la faible portée de cette technologie.

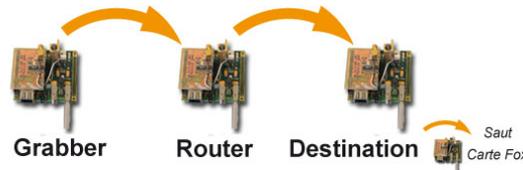


FIG. 3.5 – Notre réseau point à point

Pour la communication point à point, nous avons travaillé au niveau de la couche L2CAP de la pile Bluetooth en utilisant les Sockets. Les Sockets sont généralement utilisés pour les architectures utilisant le protocole TCP²⁴ mais les développeurs de BlueZ [110], qui est la définition de la pile Bluetooth sous Linux, ont utilisé le même principe. Le principe des Sockets, qui constitue un point de communication par lequel une application envoie ou reçoit des paquets dans un réseau, est simple et distingue deux principales entités : un client et un serveur. Le principe de communication est le même : le serveur écoute sur un port et attend les requêtes provenant du client, sauf que la paire (*adresse IP*²⁵, *numéro de port*) devient (*adresse du dispositif Bluetooth (BADDR*²⁶), *PSM*²⁷).

Après avoir défini l'architecture réseau de notre expérience, le choix du type d'information à transférer s'est posé, en l'occurrence ici des images. Le standard de compression d'images bien connu JPEG développé par le groupe qui porte son nom et dont l'acronyme est *Joint Photographic Expert Group* [111] a été choisi. JPEG est un protocole peu propice aux analyses automatiques d'images car sa compression se fait à perte, avec un taux de perte ajustable, mais il offre plusieurs avantages :

- une implémentation existante prête à l'emploi,
- une complexité algorithmique tirant avantage de la puissance de la carte Fox,
- un compromis qualité/taille de l'image que nous avons conservée fixe dans notre expérience mais qui pourrait s'adapter aux conditions énergétique sur le réseau.

Un autre standard bien connu développé par le même groupe est JPEG 2000. JPEG et JPEG 2000 diffèrent sur des aspects comme l'efficacité de compression et la complexité de la scalabilité. JPEG 2000 est une norme de compression d'images basée sur la transformation d'ondelettes qui offre de modestes améliorations par rapport à JPEG. Son principal avantage est la grande souplesse de son code. Le codage de JPEG 2000 a une complexité de calcul importante parce qu'il utilise des blocs de codage arithmétiques intégrés (*embedded arithmetic block coding*) avec un algorithme de troncature optimisé [112] tandis que JPEG est basée sur la méthode *Discrete*

²⁴Transmission Control Protocol

²⁵Internet Protocol

²⁶Bluetooth device Address

²⁷Protocole Service Multiplexer

Cosine Transform (DCT) et utilise le codage de Huffman.

Notre choix a ainsi penché sur JPEG qui est un format accessible depuis notre webcam. Par contre, JPEG 2000 de par son codage est plus tolérant aux erreurs de bits lors du transport d'images dans un réseau et règle le problème de paquets perdus.

Plusieurs dispositifs disponibles sur le marché sont capables de fournir des images de type JPEG. Le tableau 3.1 donne ces outils et ceux utilisés par les différents projets de réseaux de capteurs qui utilisent du multimedia. On constate très bien d'après les résultats figurant sur ce tableau, le compromis entre la qualité de la caméra et la consommation énergétique. En effet, la petite caméra Cyclops qui est de faible résolution (352x288 à 10 fps²⁸) consomme 46 mW tandis que la caméra PTZ de haute résolution (1024x768 à plus de 30 fps) et directionnelle consomme 1W. Nous avons choisi une solution intermédiaire : la Labtec Webcam Pro (ou la Logitech webcam).

Plateforme	Camera	Consommation
Crossbow mote, SensEye	Cyclops	46 mW
Crossbow mote, SensEye	CMUcam	200 mW
Fox	The Labtec Webcam Pro	200 mW
Fox, Panoptes, SensEye	Logitech webcam	200 mW
SensEye	PTZ camera	1W

TAB. 3.1 – Consommation des caméras [103]

3.2.4 Résultats expérimentaux

Dans cette partie, nous allons décrire et présenter les résultats d'expérimentations en environnement réel. Comme décrit précédemment, notre réseau est formé de trois nœuds principaux, un qui compresse des images JPEG puis les envoie appelé *grabber*, et un nœud qui sert de rélai jusqu'à une station base : le *router*. Il existe plusieurs classes de dispositifs Bluetooth mais ici nous utilisons la classe I et la classe II respectivement capable de transmettre à 100 mètres et à 10 mètres. Le *grabber* peut être équipé d'une webcam de type Labtec Webcam Pro (ou Logitech QuickCam Zoom (Fig. 3.6)).



FIG. 3.6 – La carte Fox, une Webcam Logitech et un dongle Bluetooth

Nous allons changer le nombre de pixels respectivement 320x240 et 640x480 pendant l'expérience dans le but d'étudier le délai et la consommation des cartes pendant le transfert d'images

²⁸fps est un acronyme de frame par secondes qui signifie en français image par seconde

de différentes taille avec des dongles de type Belkin de classes I et II. La distance entre le *router* et les autres nœuds est de 4 mètres. Pour étudier la consommation énergétique durant le transfert, nous avons utilisé l'analyseur d'énergie présenté précédemment.

La Fig. 3.7 met en lumière les résultats obtenus par le *grabber* pendant la transmission d'images de taille 320x240 et 640x480 en utilisant les dongles Bluetooth de classe I et II. Nous constatons qu'il y a de nombreuses variations au niveau du débit. En effet, les algorithmes utilisés par JPEG changent la taille des images pour le même nombre de pixels. Tandis que le nombre de pixels est le même, le nombre d'octets nécessaire pour décrire ces pixels (avec la perte d'information) changent avec l'entropie de la photo. Une image prise dans le noir par exemple où il y a plusieurs pixels qui se répètent sera facile à décrire et donc plus rapide à transférer. Pour des images de 320x240 et 640x480 pixels, leurs tailles varient respectivement entre 5197 - 11445 octets et 23287 - 36231 octets.

Alors que la littérature définit le débit théorique optimal de Bluetooth autour de 1 Mbps jusqu'à 3 Mbps, nous constatons que dans nos conditions expérimentales, il est compris entre 0,16 Mbps (soit 0,02 Mo/s) et 1,2 Mbps (soit 0,15 Mo/s). Les meilleurs débits sont atteints pendant le transfert d'images de 320x240 pixels avec les dongles de classe II (0,96 Mbps) et de classe I (1,2 Mbps) qui représentent respectivement 32% et 40% du débit théorique maximal.

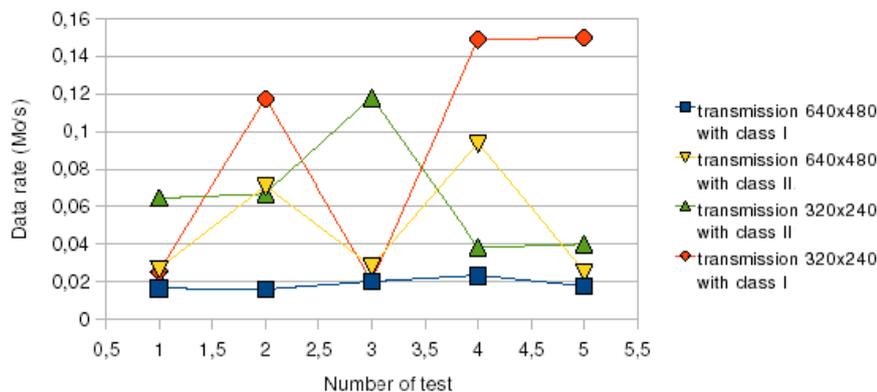


FIG. 3.7 – Débit du grabber

La Fig. 3.8 et la Fig. 3.9 montrent le débit obtenu par le *router* pendant l'étape de réception/transmission d'images de taille 320x240 et 640x480 pixels en utilisant respectivement des dongles Bluetooth de classe I et de classe II. Dans chaque graphe, le débit de transmission est meilleur que celui de réception même si il existe un point sur la Fig. 3.9 avec la classe II où la transmission et la réception ont le même débit autour de 0,12 Mbps.

Le débit pendant la réception est toujours entre 0,08 Mbps et 0,16 Mbps sur chacun des graphes tandis que le meilleur résultat est 0,56 Mbps et 0,96 Mbps respectivement pour la classe II et la classe I. Ces résultats ont été réalisés avec les images de taille 320x240 pixel. Ainsi, nous pouvons conclure que le dongle Bluetooth de classe I fournit de meilleurs résultats que celui de classe II.

Cependant, sur la Fig. 3.8, le débit de transmission pour les images de taille 320x240 et

640x480 pixels est en moyenne le même autour de 0,72 Mbps. Nous pouvons conclure que la classe I est moins dépendante de la taille des images que la classe II. Les meilleurs résultats obtenus avec la classe I représentent 32% du débit théorique optimal (3Mbps) tandis qu'il est égal à 19% pour la classe II. Les meilleurs débits pour le *grabber* sont 1,2 Mbps et 0,96 Mbps respectivement pour la classe I et la classe II tandis que pour le *router*, on obtient 0,96 Mbps pour la classe I et 0,56 Mbps pour la class II.

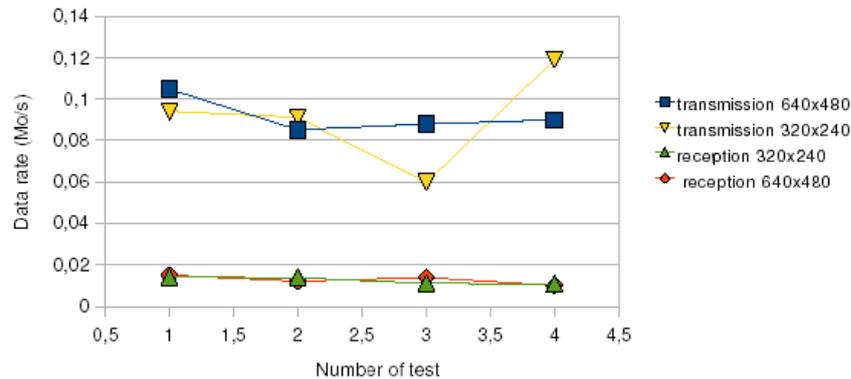


FIG. 3.8 – Débit du router en utilisant le dongle de class I

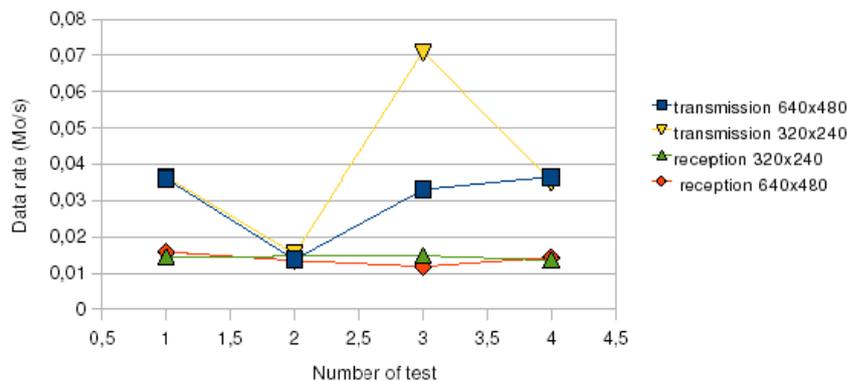


FIG. 3.9 – Débit du router en utilisant le dongle de class II

Les expériences qui précèdent montrent que le dongle de class I fournit de meilleurs résultats mais que se passe-t-il au niveau de la consommation énergétique ?

Le tableau 3.2 illustre la consommation du *router* respectivement en utilisant le dongle de classe I et de classe II. La consommation est difficile à analyser car nous n'avons accès qu'au niveau haut du protocole logiciel (pas de contrôle matériel sur le dongle) est donc notre seul mode d'analyse est la consommation globale de la carte, somme de la consommation de la carte Fox et de ses périphériques, incluant le dongle USB Bluetooth.

Les consommations des différents dongles sont très proches. Cependant, il existe une petite différence d'environ 20 mW pour la classe I et 10 mW pour la classe II. En effet, la transmission

d'images de taille 640x480 et 320x240 avec la classe I consomme en moyenne respectivement 0,61 W et 0,59 W tandis que pour la classe II la consommation moyenne est 0,57 W et 0,58 W.

De ce fait, nous remarquons que pour des dispositifs Bluetooth de classe I et de classe II, le router utilise presque la même consommation pendant la réception et la transmission d'images de tailles différentes. Néanmoins, la classe I consomme 30 mW en plus que la classe II.

Nombre de pixels / classe du dongle	Classe I	Classe II
320x240	0,59 W	0,58 W
640x480	0,61 W	0,57 W

TAB. 3.2 – Consommation du router

Le tableau 3.3 décrit la consommation énergétique du *grabber* respectivement en utilisant la classe I et la classe II pendant l'acquisition d'une image et sa transmission. La classe I utilise en moyenne 3,14 W et 3,13 W respectivement pour l'acquisition d'images de résolution 640x480 et 320x240 et leurs transmissions respectives. Il y a une différence d'environ 10 mW. L'observation de ce qui précède, d'une part l'étude de la compression JPEG puis la transmission et d'autre part le routage d'une image c'est-à-dire sa réception et sa retransmission, montre que l'activité la plus coûteuse en énergie est l'acquisition d'images. En effet, l'activité du *Router* représente en moyenne seulement 19% de la consommation du *Grabber* pour la classe I. Il faut aussi noter qu'au niveau du *Router* aucune stratégie de mise en veille du processeur n'est utilisée.

Nous constatons que la consommation du *Grabber* quand il utilise le dongle de classe II est la même en moyenne autour de 3,13 W.

Nombre de pixels / classe du dongle	Classe I	Classe II
320x240	3,13 W	3,13 W
640x480	3,14 W	3,13 W

TAB. 3.3 – Consommation du grabber

3.3 Synthèse sur le niveau supérieur de notre architecture

Dans cette partie nous avons montré les performances d'une nouvelle plateforme de réseau de capteurs de haut niveau capable de transmettre des images de bonne résolution. La souplesse de cette nouvelle plateforme, appelée la carte Fox, a permis d'ajouter différents dispositifs comme une webcam, un analyseur d'énergie et une interface Bluetooth. Nous avons étudié le débit et la consommation énergétique pendant la transmission d'images de différentes tailles en utilisant des interfaces Bluetooth de classe I et de classe II dans une simple architecture réseau point à point. Nous nous sommes focalisés sur le nœud qui capture les images et celui qui les route.

Les résultats expérimentaux montrent que pendant que le meilleur débit atteint durant l'utilisation du dongle de classe I est d'environ 32% (router) et 40% (grabber) du débit théorique optimal de Bluetooth, il est de 19% (router) et 32% (grabber) pour la classe II dans les mêmes

conditions. Nous remarquons aussi que la taille des images de type JPEG change en fonction de la cible, ce qui est dû aux algorithmes utilisés par cette norme et c'est pourquoi le débit du *grabber* change régulièrement. La consommation énergétique du *router* représente seulement 19% de l'activité du *grabber*. De ce fait, nous pouvons conclure que pendant que le dispositif Bluetooth de classe I consomme plus d'énergie que celui de classe II (environ 30mW), il offre un meilleur débit avec un rayon de communication dix fois plus grand. Ces résultats montrent le compromis entre une stabilité de transmission et la consommation énergétique.

Nous pouvons conclure que tandis que la carte Fox offre beaucoup de fonctionnalités, elle a une consommation expérimentalement validée de 1 à 3 W. Cette consommation est acceptable pour une station fixe équipée de panneaux solaires mais inacceptable pour des systèmes mobiles, de par la taille des panneaux solaires nécessaires à l'alimentaion de ce type de dispositifs, d'où l'étude dans la prochaine section d'une plateforme à faible consommation mais aussi à faible capacité de calcul par rapport à la carte Fox : la Tmote Sky.

3.4 Présentation du moyen de communication entre les différents niveaux

Dans le but de décrire progressivement l'architecture générale de notre plateforme en prenant en compte la gestion énergétique et après avoir présenté le niveau haut de cette architecture, nous allons nous intéresser aux capteurs à faible consommation qui formeront le niveau le plus bas tels que la Tmote Sky. Ensuite, nous étudierons le principal moyen de communication des plateformes de niveau bas : IEEE 802.15.4 et ses stratégies d'économie d'énergie.

3.4.1 La Tmote Sky

La Tmote Sky de dimension 32x80 mm fait partie de la famille des Telos, c'est à dire des nœuds de capteurs avec USB. Elle est fabriquée par Moteiv (actuellement Sentilla [113]) et se base sur un microcontrôleur de Texas Instruments appelé TI MSP430, spécialement conçu pour des applications embarquées à basse consommation. C'est un dispositif qui consomme très peu et est compatible avec la norme IEEE 802.15.4 pour la communication sans fil. Comme la mote Micaz son transceiver est de type Chipcon CC2420 radio. Le microcontrôleur MSP430F1611 disponible sur la Tmote Sky est cadencé à 8 MHz et se compose de 10ko SRAM, 48ko Flash et 1Mo de mémoire Flash externe. Les différentes données scalaires qu'elle est susceptible de fournir sont : l'humidité, la température et le niveau de luminosité.



FIG. 3.10 – La Tmote Sky

Après avoir défini les principaux éléments de notre architecture, nous allons nous intéresser au moyen de communication entre ces deux différents niveaux : le protocole IEEE 802.15.4. Ce standard de communication est quasiment imposé par l'architecture des plateformes de niveau

bas dans ce domaine, qui utilise par défaut cette technologie à cause de ses stratégies intrinsèques d'économies d'énergie.

3.4.2 Le protocole IEEE 802.15.4

Le protocole IEEE 802.15.4, qui constitue la couche MAC et la couche physique de ZigBee, est un protocole adapté aux réseaux de capteurs à cause de sa faible consommation énergétique. La particularité de ce standard est sa couche MAC qui offre une *Supertrame* capable de mettre le module de transmission en veille en fonction d'un schéma prédéterminé selon le mode utilisé. En effet, ce standard qui a un débit de 250 kb/s spécifie deux modes d'accès au médium : un mode non synchronisé (CSMA/CA classique) et un mode synchronisé où des noeuds dans une topologie étoile se synchronisent avec un coordinateur. Le protocole IEEE 802.15.4 offre deux types de dispositifs réseau : les FFDs (Full Function Device) ayant toutes les fonctions possibles et les RFDs (Reduced Function Device) qui sont de simples dispositifs avec des ressources modestes. Les RFDs constituent en général les feuilles du réseau car leurs ressources sont limitées, tandis qu'un FDD est une unité de traitement sans contrainte énergétique avec plusieurs fonctionnalités dont le routage.

Ces caractéristiques sont en accord avec la topologie en étoile qu'offre 802.15.4 où tous les nœuds communiquent avec un nœud central appelé coordinateur PAN. Dans cette topologie, il n'y a pas de communication entre RFDs. Cette description est conforme à notre architecture où les plateformes de bas niveau tel que la Tmote Sky sont des RFDs et la carte Fox, un FDD qui coordonne l'échange d'information avec son entourage. L'autre topologie qu'offre 802.15.4 est une architecture point à point où il n'y a aucune hiérarchie entre les nœuds (Fig. 3.11).

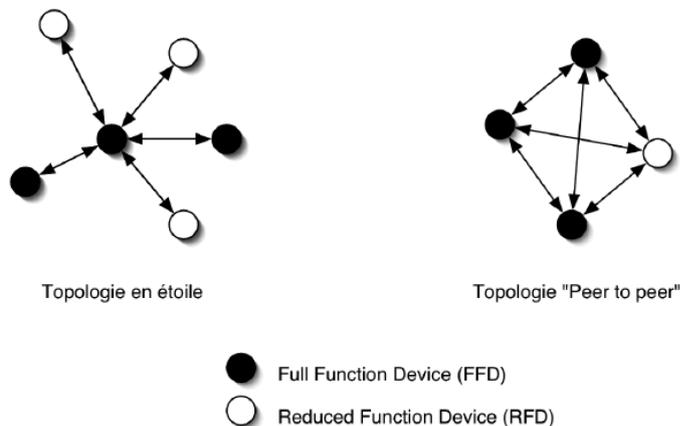


FIG. 3.11 – Topologies proposées par le standard IEEE 802.15.4 [114]

Nous allons nous focaliser sur le mode synchronisé qui offre de meilleures performance au niveau de la gestion énergétique. Dans ce mode, les communications sont garanties par un espace temporel appelé *Supertrame* (Fig. 3.12). La *Supertrame* est composée de deux principales portions : une portion active et une portion inactive. Les nœuds interagissent pendant la période active qui est divisée en 16 slots de même durée et entrent en somnolence pendant la période inactive pour économiser de l'énergie.

La portion active se compose de deux périodes : une période avec contention (noté CAP²⁹) et une période sans contention optionnelle (noté CFP³⁰). La méthode d'accès au médium avec contention correspond au protocole CSMA/CA classique tandis que dans la méthode d'accès sans contention, qui est optionnelle, les accès sont contrôlés par le coordinateur.

Aussi, cette dernière méthode offre la possibilité aux nœuds du réseau de demander un ou plusieurs slots au coordinateur. Ces slots temporels dédiés, qui sont alloués par le coordinateur lorsque le réseau le permet, porte le nom de *Guaranteed Time Slots* (GTS). Cette période constitue les derniers slots de la supertrame pendant que la période avec contention occupe les premiers slots.

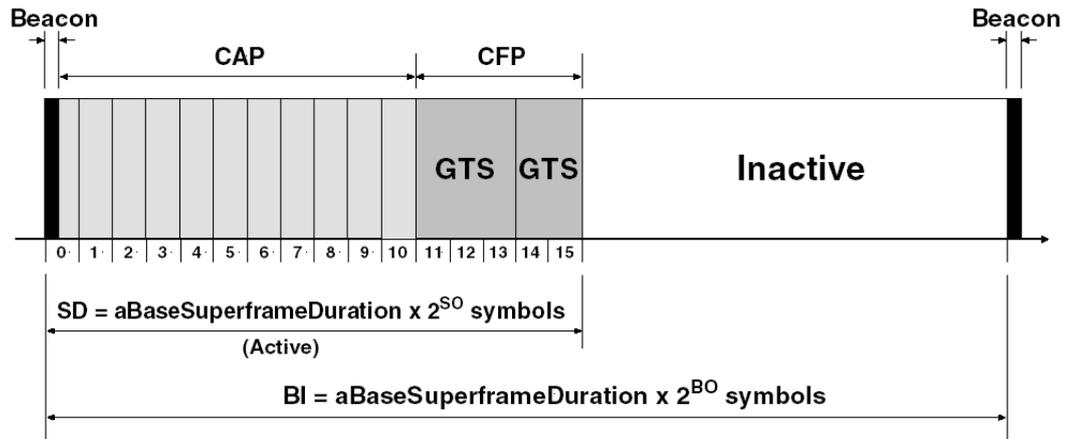


FIG. 3.12 – Structure d'une *Supertrame* [115]

Il faut noter que les demandes de GTS ou d'association au réseau se font pendant la période avec contention, ce qui veut dire qu'il faut limiter la période sans contention. Le coordinateur, qui envoie périodiquement les trames balises (appelées beacon) pour synchroniser les nœuds du réseau, détermine deux paramètres importants : *le Beacon Order (BO)* et *le Superframe Order* qui sont respectivement responsables de la durée de la supertrame (intervalle entre deux beacons noté BI³¹) et la durée de la période active noté SD³². De là, les équations permettant de déterminer les paramètres précédents sont les suivantes :

$$\begin{aligned}
 BI &= aBaseSuperframeDuration * 2^{BO}, & (1) \\
 SD &= aBaseSuperframeDuration * 2^{SO}, & (2) \\
 \text{avec } 0 \leq SO \leq BO \leq 14
 \end{aligned}$$

La question fondamentale est de savoir quel est le point optimal de la supertrame entre la portion active et la portion inactive. C'est à dire que nous cherchons un endroit sur l'axe temporel de la supertrame qui permet d'une part au nœud d'être conscient du réseau et de pouvoir échanger des informations et d'autre part de gérer sa consommation énergétique en somnolant. C'est tout simplement le compromis existant avec le standard IEEE 802.15.4 entre l'économie d'énergie et la latence impulsée par la période de somnolence, c'est pour cette raison que dans la partie qui

²⁹CAP est un acronyme anglais de Contention Access Period

³⁰CFP est un acronyme anglais de Contention Free Period en anglais

³¹BI est un acronyme de Beacon Interval

³²SD est un acronyme de Superframe Duration

suit nous allons dans un premier temps présenter notre plateforme hiérarchique et générique de réseaux de capteurs [116], puis nous allons évaluer le compromis énergie/délai du protocole IEEE 802.15.4 sur notre plateforme hiérarchique et générique de réseaux de capteurs [117].

3.5 Notre architecture hiérarchique et générique de réseaux de capteurs

Cette architecture repose sur la carte Fox qui constitue le nœud de haut niveau, concentrateur et capable de faire des tâches complexes, et sur la Tmote Sky au niveau inférieur, chargée de collecter de simples données scalaires. Nous allons donc mettre l'accent dans cette partie sur les communications entre les différents niveaux de cette plateforme hiérarchique. La Tmote Sky est par défaut compatible avec la norme IEEE 802.15.4 et utilise TinyOS comme environnement de développement. Open-ZB [118], un environnement opensource développé sous TinyOS, va fournir les primitives MAC du protocole IEEE 802.15.4 nécessaire pour communiquer avec la carte Fox qui sera équipée d'un dongle IEEE 802.15.4 [119] (Fig. 3.13). Cet environnement va nous permettre de modifier expérimentalement les paramètres de la couche MAC du protocole IEEE 802.15.4.



FIG. 3.13 – La carte Fox, la Tmote Sky et le dongle IEEE 802.15.4/ ZigBee

Cette architecture bénéficie, grâce aux choix logiciels précédents, de trois niveaux de généralité :

- Au niveau communication : La flexibilité de la carte Fox grâce à ses deux ports USBs permet de l'adapter à plusieurs technologies sans fil,
- Au niveau logiciel : Open-ZB qui est une implémentation sous TinyOS peut être adapté à différentes plateformes de bas niveau utilisant TinyOS à partir des mêmes fichiers sources,
- Au niveau application : Les deux niveaux de généralité précédents et l'architecture à deux niveaux où des tâches complexes peuvent être effectuées au niveau haut permet de l'adapter à plusieurs applications.

Plusieurs travaux de recherche ont tenté d'améliorer les performances du standard IEEE 802.15.4, certains en modifiant les paramètres des équations (1) et (2) tels que le *Beacon Order (BO)* [120], d'autres en définissant des nouveaux schémas de fragmentation de données pour des petits *Beacon Interval (BI)*, c'est à dire pour des supertrames courtes [115]. Cependant, dans [115], les résultats sont obtenus par simulations et dans [120], de manière analytique puis confirmés par simulation.

Dans ce travail, nous allons évaluer de manière expérimentale les performances de cette technologie sur une plateforme hiérarchique en modifiant les paramètres des équations (1) et (2), puis nous allons confronter les résultats expérimentaux aux assertions théoriques.

3.5.1 Implémentation du protocole IEEE 802.15.4 sur notre plateforme

Dans cette section, nous allons décrire les résultats obtenus pendant la transmission de données à l'aide du protocole IEEE 802.15.4. Nous allons mettre l'accent sur la consommation énergétique et sur le délai lorsque nous allons modifier successivement les paramètres de la Superframe grâce aux équations (1) et (2). La Tableau 3.4 illustre la consommation énergétique de la carte Fox, coordinateur PAN, mesurée à l'aide de son analyseur d'énergie intégré quand elle reçoit d'une part, et transmet d'autre part des données tests de 4 octets de manière continue, c'est à dire sans aucune période de somnolence.

Inactive	Mode veille	Transmission	Réception
0,56 W	0,56 W	0,63 W	0,69 W

TAB. 3.4 – Consommation énergétique de la carte Fox selon différents modes

Cette expérience a été réalisée afin d'estimer la consommation de la carte dans différents modes mais aussi pour évaluer le gain énergétique lorsqu'un dispositif est en mode somnolence car la mise en mode veille reste une procédure bien connue pour économiser l'énergie. D'après ce tableau, nous constatons non seulement que la réception consomme plus que la transmission mais aussi que la mise en veille du transceiver consomme autant que la carte lorsqu'elle ne fait aucune activité, ce qui confirme le poids du module de transmission au niveau de la consommation.

Cependant, l'utilisation du mode somnolence pendant longtemps entraîne une inactivité du nœud pendant une période, et donc une latence du réseau, c'est pour cette raison que nous allons modifier les paramètres des équations (1) et (2) pour observer la consommation de la carte Fox avec une très courte période de somnolence. Nous ajustons le beacon order (BO) à 6 et le Superframe Order SO à 4, ce qui fait en appliquant (1) et (2), une période d'inactivité de 0,74 s (BI-SD). En effet, en considérant (1) et (2) :

$$BI = aBaseSuperframeDuration * 2^{BO}, \quad (1)$$

$$SD = aBaseSuperframeDuration * 2^{SO}, \quad (2)$$

avec $aBaseSuperframeDuration = aBaseSlotDuration * aNumSuperframeSlots$

Or $aBaseSlotDuration = 60$ symboles et $aNumSuperframeSlots = 16$ d'où
 $aBaseSuperframeDuration = 60 * 16 = 960$ symboles

Cependant, $SymbolDuration = 16$ microsecondes soit $aBaseSuperframeDuration = 960 * 16$ microsecondes = $960 * 0,016$ millisecondes = $15,36$ ms.

Les équations (1) et (2) se traduisent donc de la manière suivante :

$$BI = 15,36 * 2^{BO}, \quad (1)$$

$$SD = 15,36 * 2^{SO}, \quad (2)$$

Avec $BO = 6$ et $SO = 4$, nous obtenons : $BI = 15,36 * 2^6 = 983,04$ ms,

et $SD = 15,36 * 2^4 = 245,76$ ms, soit la Période inactive : $BI-SD = 983,04 - 245,76 = 0,737$ s.

La Fig.3.14 détaille la consommation énergétique de la carte pour cette période d'inactivité inférieure à une seconde avec une interrogation de l'analyseur d'énergie à 9600 bauds.

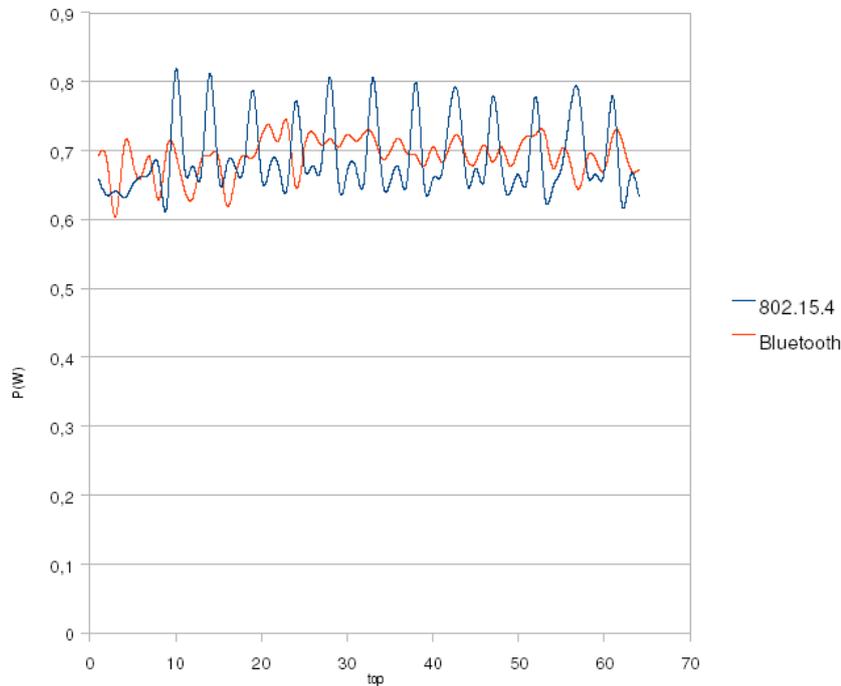


FIG. 3.14 – Consommation énergétique de la carte Fox une courte période d'inactivité de 0,73 s

Sur cette figure, nous comparons la consommation énergétique de IEEE 802.15.4 et Bluetooth quand la carte Fox reçoit de manière continue un flux de données de 4 octets de la Tmote Sky d'une part et d'un ordinateur portable d'autre part en utilisant respectivement, IEEE 802.15.4 et Bluetooth. Le premier niveau de généricité a permis d'adapter la carte Fox à différentes technologies de communication. La comparaison entre la consommation de Bluetooth et IEEE 802.15.4 (Fig.3.14) montre que pendant que Bluetooth a une consommation constante autour de 0,7 W, IEEE 802.15.4 présente des pics jusqu'à 0,8 W et chute ensuite rapidement autour de 0,6 W. La consommation moyenne de ces deux technologies pendant cette expérience est la même soit d'environ 0,7 W.

Ces résultats montrent qu'une transmission continue de données en utilisant IEEE 802.15.4 n'est pas plus satisfaisante que Bluetooth. Nous pouvons conclure en disant que ce n'est pas seulement le choix de la norme IEEE 802.15.4 qui permet de consommer moins mais aussi le choix des différents schémas de veille. En effet, le choix d'une bonne stratégie de mise en veille grâce aux paramètres BO et SO est très important. Dans cette expérience, nous avons changé de manière continue les modes de la carte avec une très courte période de somnolence, ce qui n'a pas montré les avantages de IEEE 802.15.4 par rapport à Bluetooth même si la consommation du standard 802.15.4 descendait jusqu'à 0,6 W pendant la période de veille mais en moyenne elle est quasiment la même. Avec ces premiers résultats où la période de veille est courte n'engendrant aucun gain énergétique comparé à Bluetooth, nous allons étendre la période d'inactivité et nous allons observer les variations au niveau du délai.

Dans les prochaines expériences, nous allons utiliser le second niveau de généricité de notre plateforme d'une part pour valider ce second niveau et d'autre part parce qu'il est plus facile

d'intégrer un analyseur d'énergie à la mote MicaZ qu'à la Tmote Sky qui est compacte. La MicaZ et la Tmote Sky utilisent le même transceiver : un Chipcon CC2420.

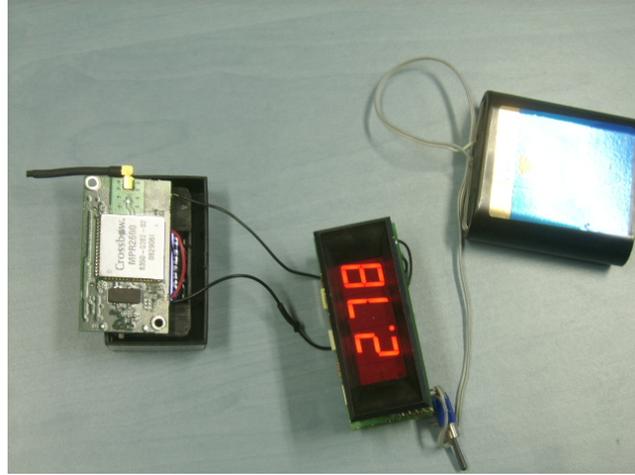


FIG. 3.15 – La mote MicaZ équipée d'un ampèremètre commercial

Ainsi, le tableau 3.5 compare respectivement les consommations énergétique de la carte Fox et de la mote MicaZ pendant la transmission de données de 4 octets vers un PC en mode continu sans mise en veille et avec une période de somnolence de 1,7 s (BO = 7, SO = 4).

Les variations sont similaires : la période de veille permet au dispositif d'économiser de l'énergie. La consommation de la carte Fox sans période de sommeil est d'environ 0,65 W tandis qu'avec une période de somnolence de 1,7 s est d'environ 0.6 W (tableau 3.5), ce qui constitue un gain de 0.05 W et représente 8% de gain comparé à la transmission sans période de somnolence. La consommation de la micaZ sans aucune mise en veille est de 9.93 mW pendant qu'elle avoisine 9.43 mW avec une période de veille de 1,7 s (tableau 3.5), soit un gain de 0,5 mW qui représente 5%. Cette période de veille qui a été étendue a permis de mieux distinguer la consommation dans chaque état pour chacune des cartes.

Consommation / Plateforme	Carte Fox	MicaZ
Mode actif	0,65 W	9,93 mW
Période de veille de 1,7 s	0,6 W	9,43 mW
Gain énergétique	8%	5%

TAB. 3.5 – Evaluation énergétique des différentes plateformes

Ces résultats servent de support pour l'estimation de la différence de consommation entre la carte Fox et la mote. En effet, pendant la transmission en continu, nous constatons que la carte Fox consomme 0,65 W pendant que la mote utilise environ 9.93 mW, ce qui traduit la différence de consommation de la mote qui est 60 fois moins que celle de la carte Fox. De ce fait, nous pouvons conclure que la mote utilise moins de 2% de la consommation de la carte Fox pour la même activité. Aussi, pour la même période de somnolence, le gain énegetique de la carte Fox est de 8% pendant que pour la mote, il est de 5%. Une période de somnolence de 1.7 s est 3% plus bénéfique pour la carte Fox que pour la mote.

Comme mentionné précédemment, pendant les transmissions successives, nous nous sommes intéressés aux délais sur les différentes plateformes afin d'étudier le compromis entre l'économie d'énergie et le délai dans le protocole IEEE 802.15.4.

Les Fig.3.16 et Fig.3.17 montrent le délai obtenu pendant la transmission de données respectivement sans et avec une période de veille.

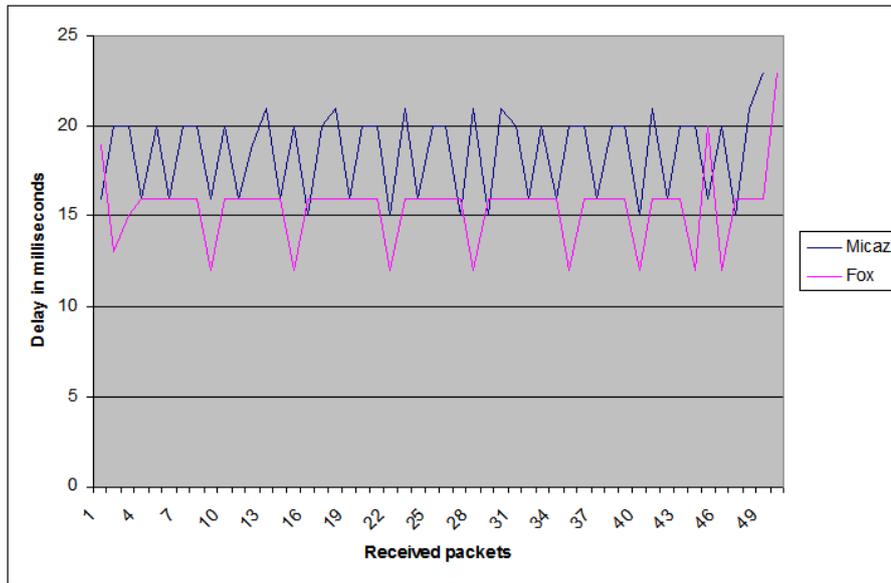


FIG. 3.16 – Délai de la carte Fox et de la MicaZ sans période de somnolence

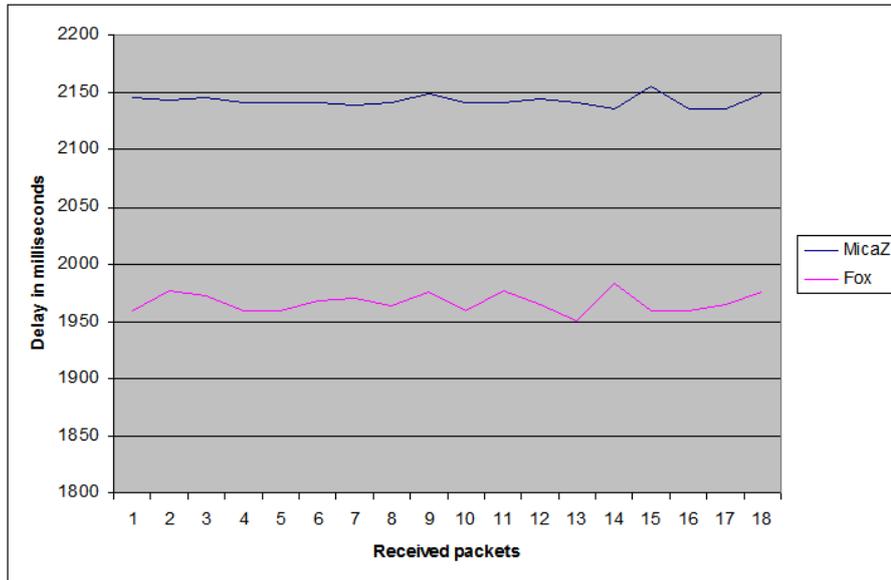


FIG. 3.17 – Délai de la carte Fox et de la MicaZ avec une période de somnolence de 1,7 s

Nous constatons que la carte Fox a besoin d'environ 15.56 ms pour transmettre des données sans période de somnolence et d'à peu près 1966 ms pour une période de somnolence de 1.7 s, soit

une différence de 1951 ms. Par contre, la MicaZ utilise 18.58 ms pour transmettre des données sans période d'activité et 2138 ms pour une période de somnolence de 1,7 s, soit une différence de 2120 ms.

Ces résultats sont conformes aux résultats théoriques qui montrent que pour un SO minimal, nous avons 15,36 ms qui est le temps entre deux cycles minimal. En effet, en prenant (2) avec $SO = 0$ on obtient : (2) $SD = 15,36 * 2^0$, soit $SD = 15,36$ ms ce qui constitue le temps minimal entre deux cycles. C'est pour cette raison que dans [121], les auteurs disent qu'avec la norme IEEE 802.15.4, il est impossible de satisfaire des applications industrielles nécessitant des temps de cycles inférieurs à 15,36 ms. Si un nœud doit envoyer un message au coordinateur, il peut le faire dans les meilleurs des cas que toutes les 15,36 ms, ce qui est proche des résultats expérimentaux trouvés.

L'étude comparative entre la carte Fox et la mote MicaZ pendant le transfert de données sans période de somnolence montre que les deux cartes utilisent quasiment le même temps pour transmettre même si la carte Fox est légèrement plus rapide. La Fig.3.17 montre que le module de transmission de la carte Fox se ré-active et transmet plus rapidement que la mote, nous avons remarqué un écart de 172 ms entre les délais de la carte Fox et de la mote après une période de somnolence de 1,7 s. Nous constatons que la carte Fox est plus réactive après une période de somnolence, elle a besoin de 0.26 s (1.96 - 1.7) pour se réveiller et transmettre une donnée tandis que la MicaZ utilise 0.43 s (2.13 - 1.7).

Les expériences précédentes mettent en lumière les avantages de la carte Fox par rapport à la mote MicaZ, puisque malgré la consommation d'énergie de la mote qui ne représente que 2% de celle de la Fox, la Fox réagit mieux à une période de somnolence avec un gain énergétique de 8% et une réactivité de 0,26 s. Ces résultats vont nous permettre de déterminer les coefficients nécessaires à l'évaluation du compromis énergie/délai de la norme IEEE 802.15.4 sur notre plateforme hybride de réseau de capteurs.

3.5.2 Evaluation du compromis énergie/délai du protocole IEEE 802.15.4

Nous allons dans un premier temps observer les tableaux récapitulatifs des expériences de la section précédentes (tableau 3.6 et 3.7). Ces résultats vont nous fournir les paramètres nécessaires à l'évaluation du compromis énergie/délai du protocole IEEE 802.15.4.

Consommation / Plateforme	Carte Fox	MicaZ
Mode actif	0,65 W	9,93 mW
Période de veille de 1,7 s	0,6 W	9,43 mW

TAB. 3.6 – Evaluation énergétique des différentes plateformes

Délai / Plateforme	Carte Fox	MicaZ
Mode actif	15,6 ms	18,6 ms
Période inactive de 1,7 s	1967 ms	2139 ms
Retransmission	260 ms	430 ms

TAB. 3.7 – Evaluation des délais de transmission des différentes plateformes

Nous obtenons d'après les tableaux 3.6 et 3.7 :

- **Pour la carte Fox** : $P_{Sleep} = 0,05$ W et $P_{Actif} = 0,65$ W avec $T_{Sleep} = 0,26$ s et $T_{Actif} = 15,6$ ms.
- **Pour la mote MicaZ** : $P_{Sleep} = 0,5$ mW et $P_{Actif} = 9,93$ mW avec $T_{Sleep} = 0,43$ s et $T_{Actif} = 18,6$ ms.

Avec les coefficients ci-dessus nous pouvons déterminer l'allure de la courbe représentant la variation de puissance totale (P_{Total}) en fonction du temps de veille :

$$(3) P_{Total} = \frac{P_{Sleep} * T_{Sleep} + P_{Actif} * T_{Actif}}{T_{Sleep} + T_{Actif}}$$

La Fig. 3.18, obtenue à partir de la formule (3), illustre la variation de la puissance de la carte Fox en fonction du temps. Cette figure met en valeur le compromis fort entre l'énergie et le délai.

En effet, l'énergie consommée décroît quand le temps de veille du module de transmission croît. La grande question est donc de savoir comment trouver un point optimum entre ces deux critères qui évoluent de manière antagonistes. Une tentative de résolution de ce problème est l'introduction de la notion d' α -pénalité.

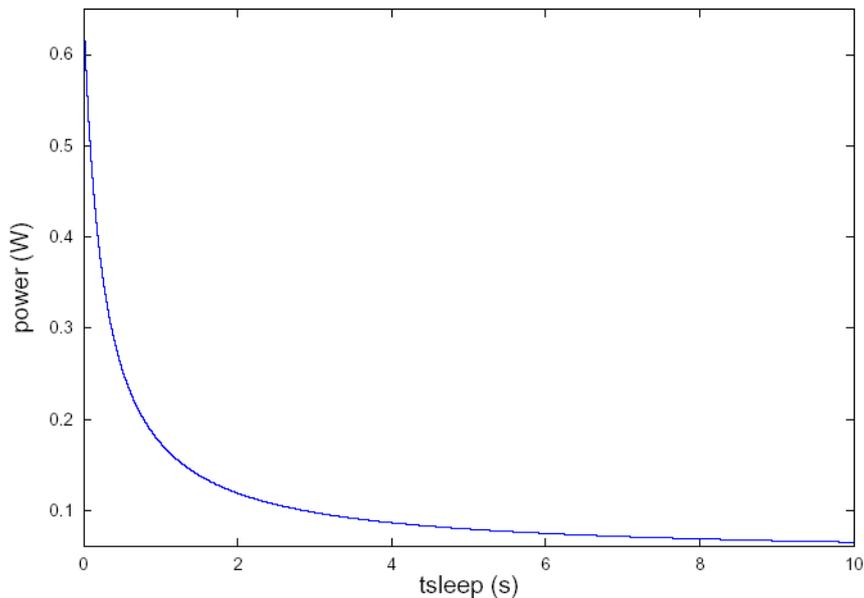


FIG. 3.18 – Evaluation de la puissance totale de la carte Fox en fonction du temps de veille

3.5.3 Introduction de l' α -pénalité

Mise en contexte

Toutes les applications de réseaux de capteurs ne répondent pas aux mêmes contraintes, certaines sont dirigées par événements et d'autres basées sur des observations. Contrairement aux applications dirigées par événements, celles basées sur des observations ne nécessitent pas de traitements en temps réel. Cependant, la stratégie utilisée pour économiser de l'énergie à l'aide de la technologie sans fil IEEE 802.15.4 est la même : mettre le module de transmission en veille.

La question de trouver un point optimum entre la mise en veille et la disponibilité dans le réseau justifie l'utilisation de l' α -pénalité. En effet, pour des applications dirigées par événements par exemple, il est important d'envoyer une information le plus rapidement possible.

L' α -pénalité offre donc la possibilité de définir théoriquement pour une application un poids qui signifie qu'être en veille trop longtemps a un coût, ce qui va nous permettre de déterminer un temps de veille optimum.

Présentation

Dans cette partie, nous allons traduire le fait qu'il n'est pas judicieux pour un module de transmission de passer essentiellement son temps en mode veille car il est coupé du réseau. Nous introduisons donc un poids qu'on appelle une pénalité relative à ce temps de veille et qui signifie que somnoler trop longtemps constitue un coût pour le réseau. Le choix de cette pénalité va nous pousser à définir un paramètre α correspondant à la pénalité et que nous allons faire varier pour voir l'allure générale de notre puissance totale. Il faut noter que le paramètre α est d'autant plus grand que la latence pénalise les performances du réseau. Nous allons ajouter à l'équation (3), le paramètre α qui se traduit de la manière suivante :

$$(4) \quad P_{Total} = \frac{P_{Sleep} * T_{Sleep} + P_{Actif} * T_{Actif}}{T_{Sleep} + T_{Actif}} + \alpha * (T_{Sleep} + T_{Actif})$$

La première étape consiste à observer la nouvelle allure de la puissance totale et de faire varier α entre 0 et 0,1 avec un pas de 0,01 et observer le faisceau de courbes résultant. Il faut noter que nous avons tenté en vain de définir physiquement à quoi rattacher la valeur α ce qui fait que la solution que nous avons choisie consiste à raisonner à partir d'un faisceau de courbes et de ne pas s'arrêter sur un paramètre physique. Nous pouvons définir α comme un paramètre qui illustre les effets de la latence de transmission de l'information sur le contrôle du phénomène observé (par exemple pour la détection de feu de forêt). La Fig. 3.19 détaille les variations de la puissance totale de la carte avec les différentes valeurs du paramètre α .

La Fig. 3.20 met en lumière la même expérience avec les paramètres de la mote MicaZ en prenant en compte le même intervalle pour le paramètre α entre 0 et 0,1 avec des pas de 0,01.

La comparaison des Fig. 3.19 et 3.20 montre non seulement qu'il y a une grande différence de consommation entre les deux plateformes dont les courbes sont sur la même échelle mais aussi que la mote réagit plus rapidement à l' α -pénalité, c'est-à-dire que lorsque le paramètre α augmente, la puissance consommée de la mote croît rapidement. De ce fait, nous allons déduire de ces courbes les temps de veille optimums rattachés aux différentes pénalités.

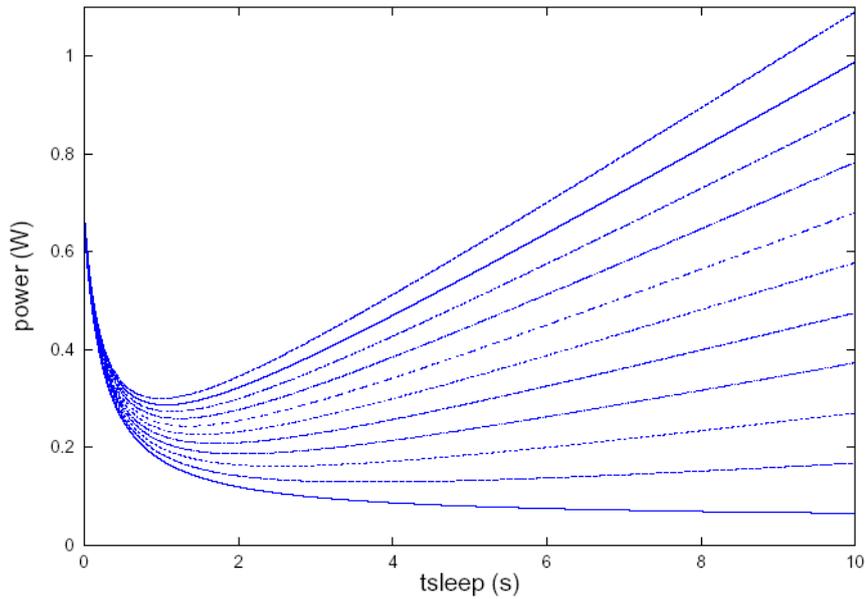


FIG. 3.19 – Evaluation de la puissance totale de la carte Fox en prenant en compte l' α -pénalité

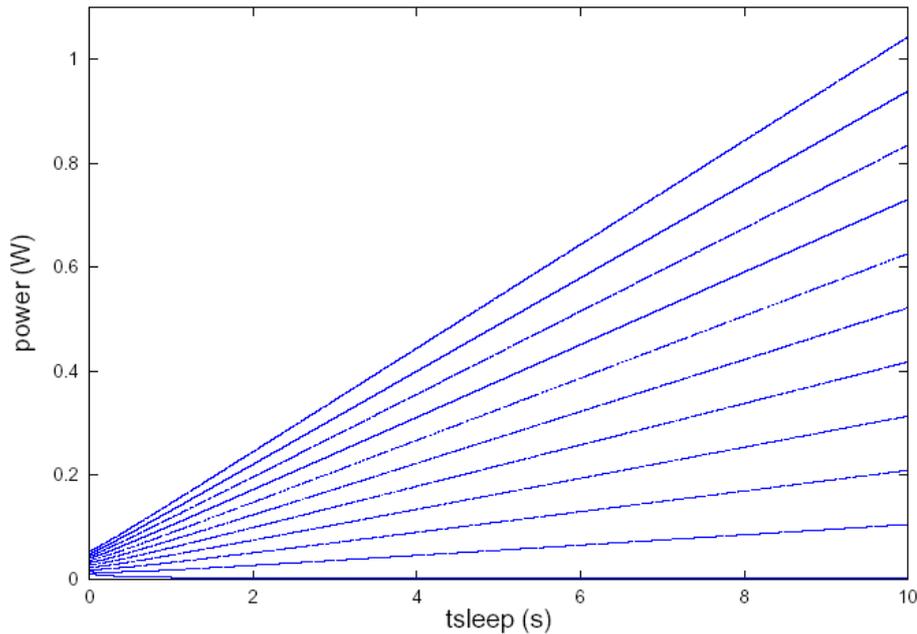


FIG. 3.20 – Evaluation de la puissance totale de la MicaZ en prenant en compte l' α -pénalité

Les Figures 3.21 et 3.22 qui identifient les minima de chaque courbe des faisceaux des Fig. 3.19 et 3.20 montrent que la mote utilise un temps très court pour économiser de l'énergie et rapidement son temps de veille devient négligeable lorsque α évolue tandis que la carte Fox a besoin de plus de temps et ne supporte pas les temps de veille trop court. Le système Linux de la plateforme de haut niveau semble être responsable de cette contrainte car elle n'est pas destinée

à la base pour la gestion d'énergie.

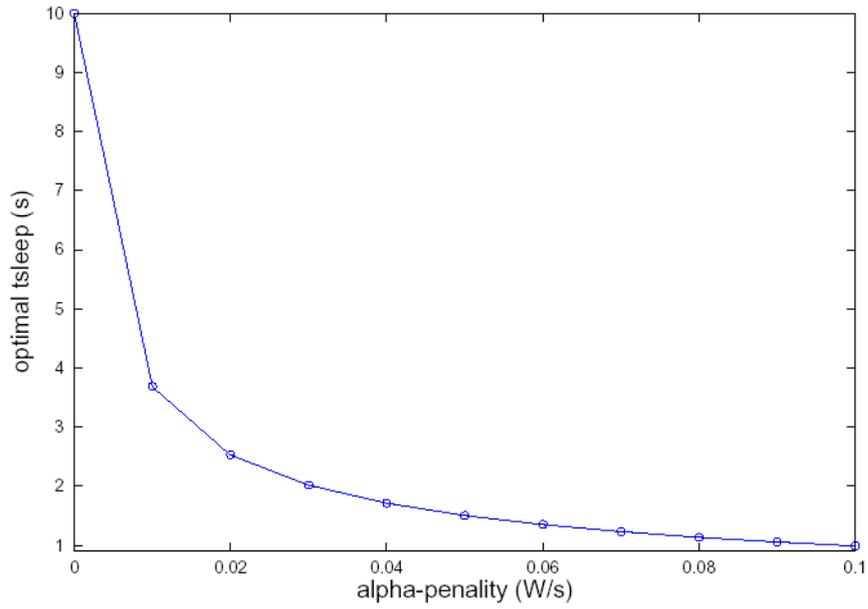


FIG. 3.21 – Variation des temps de veille optimums de la carte Fox en fonction de l' α -pénalité

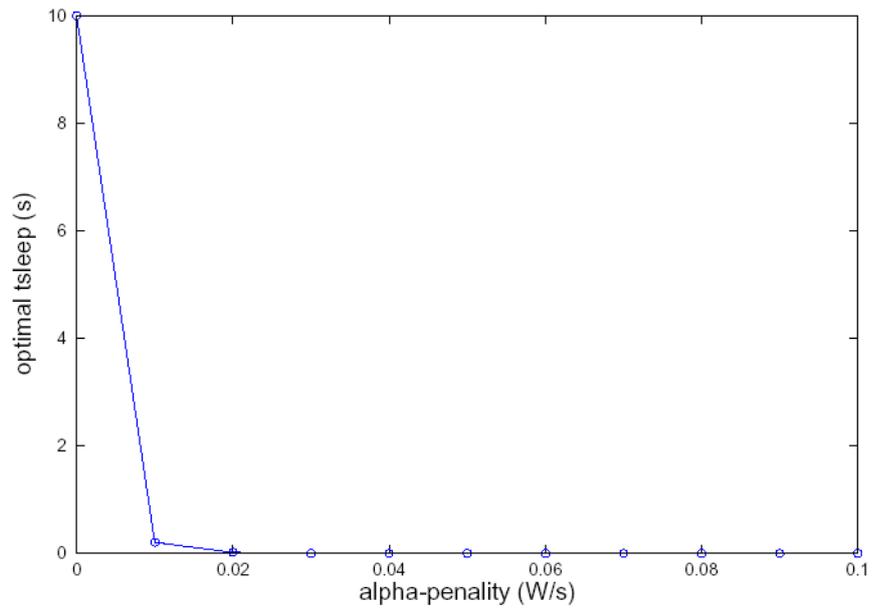


FIG. 3.22 – Variation des temps de veille optimums de la MicaZ en fonction de l' α -pénalité

3.6 Conclusion

La plateforme de haut niveau, qui utilise Linux, est non seulement plus coûteuse énergétiquement mais nécessite également un temps de veille considérable pour économiser de l'énergie par rapport aux nœuds de bas niveau qui sont plus flexibles aux temps courts. Cette contrainte fait que les nœuds de bas niveau qui ont besoin de temps très courts de veille pour économiser l'énergie et donc sont toujours prêts à envoyer des informations, obligent le nœud concentrateur à être toujours actif. Ce qui va nous pousser à développer des techniques d'économies d'énergie au niveau bas de notre architecture.

Il faut aussi noter que le compromis énergie/délai intrinsèque aux stratégies d'économie d'énergie de 802.15.4 est fort, ce qui fait qu'il n'est pas adapté aux applications avec des contraintes fortes de délai telles que les applications industrielles [121]. Dans le meilleur des cas, il faut 15,36 ms pour chaque cycle ce qui peut être une contrainte rédhibitoire pour une application où une microseconde de précision est importante.

En résumé, les conclusions de notre étude en utilisant l'alpha-pénalité montrent que :

1. Pour économiser de l'énergie les nœuds ont besoin de temps plus courts par rapport aux nœuds de niveau haut sous Linux. Ce qui valide théoriquement le fait que le niveau haut doit être alimenté en permanence car les nœuds sont susceptibles d'envoyer des informations n'importe quand et à cela se rajoute son rôle de concentrateur.
2. Lorsque le paramètre alpha croît, c'est à dire lorsqu'il y a de fortes contraintes temporelles liées à l'application, le temps optimal de veille des nœuds devient rapidement nul, ce qui signifie qu'il n'y a aucune économie d'énergie. Autrement dit, le compromis énergie/délai lié à IEEE 802.15.4 étant fort lorsque le délai devient un critère important, ce protocole n'est pas adapté [121] et l'économie d'énergie n'est pas possible.

Le premier point précédent met en évidence l'importance de définir des stratégies d'économies d'énergie au niveau des nœuds car le nœud haut doit être alimenté en permanence.

Le second point confirme l'intérêt dans ce travail pour les applications d'observation par rapport à celles contraintes par le temps d'où la compression et la stockage de données qui se placent dans un contexte sans contraintes temporelles.

Ainsi, nous allons nous focaliser dans le chapitre suivant sur du monitoring centré sur des données. Nous allons développer une stratégie d'économie d'énergie rattachée à ce type d'applications, à savoir : la compression de données.

Nouvelles techniques de compression de données dans les réseaux de capteurs

"Rien ne se perd, rien ne se crée, tout se transforme." Anaxagore de Clazomènes

Sommaire

4.1	Les algorithmes de compression	91
4.1.1	Les algorithmes avec pertes	91
4.1.2	Les algorithmes sans pertes	92
4.2	L'algorithme S-LZW	93
4.3	L'algorithme RLE (Run-Length Encoding)	94
4.4	Première contribution : l'algorithme K-RLE (K-Run-Length Encoding)	96
4.4.1	Principe	96
4.4.2	Résultats expérimentaux	98
4.5	Synthèse sur l'algorithme K-RLE	111
4.6	Deuxième contribution : l'algorithme Aganmâ	112
4.6.1	Principe	112
4.6.2	Résultats expérimentaux	114
4.7	Synthèse sur l'algorithme Aganmâ	116
4.8	Conclusion	117

La notion de réseau se rattache à la circulation et au partage d'informations par plusieurs entités interconnectées. L'évolution des réseaux tels que Internet, le réseau par excellence, ou encore les réseaux de capteurs, réseau dense par définition, montrent que la quantité de données échangée devient de plus en plus conséquente. Pour augmenter les performances d'un réseau, c'est à dire éviter son encombrement, garantir la rapidité du transfert des données ou permettre des envois simultanés, la transformation des informations est nécessaire d'où l'utilisation de techniques de compression de données. Comme le stipule l'adage "*Rien ne se perd, rien ne se crée, tout se transforme*", la compression consiste à transformer les données de sorte que la taille physique des blocs d'information soit réduite, ce qui facilite leur stockage ou encore leur transport sur un réseau jusqu'à un destinataire donné. Ce destinataire, une fois la quantité de données compressées reçue, doit les décompresser. Si les données décompressées sont identiques aux données avant compression on parle de compression sans pertes sinon, si les données ont subi des transformations, on parle de compression avec pertes.

La compression de données constitue un nouvel axe de recherche pour les réseaux de capteurs. En effet, comme mentionné dans [19, 122], la principale source de consommation énergétique pour un capteur reste dans la grande majorité des cas le transceiver, c'est pour cette raison qu'une stratégie pour économiser de l'énergie est de l'éteindre temporairement (mode somnolence) en accord avec un schéma de réveil bien précis lié à l'application. Cependant, la mise en veille du transceiver réduit le nombre de communications radio puisque le dispositif est indisponible. On peut alors se demander comment garantir le même taux de données envoyé à la station de base en réduisant le nombre de transmissions ?

Une tentative de réponse à cette question est l'utilisation de la compression de données. En effet, les nœuds de niveau bas doivent économiser de l'énergie et la compression de données peut constituer une bonne alternative. Cependant, parler de compression dans les RdCs suppose que nous ne sommes pas dans le cas d'applications dirigées par événements où le délai est une contrainte, mais plutôt sur l'observation d'un environnement et la collecte de données sans contraintes temporelles.

Un exemple simple de compression de données de tous les jours est l'envoi de SMS (Short Message Service) où les mots sont plus courts en gardant un sens facile à lire par le destinataire, le but étant d'écrire le plus rapidement possible et d'envoyer le minimum de messages car chaque message a une taille prédéfinie et un coût. Ex : *Je vais dehors* devient *J v 2Or*. Nous sommes passés de 14 caractères, espaces compris, à 7 caractères soit un gain de 50%.

Les algorithmes de compression avec ou sans pertes sont multiples car ces techniques sont de plus en plus répandues avec le développement des réseaux. Cependant, les algorithmes classiques, de par leur complexité et les faibles ressources des capteurs, ne peuvent pas être utilisés directement d'où leur adaptation aux contraintes des plateformes existantes.

Dans le travail qui suit, nous allons présenter les principaux algorithmes de compression, leurs limites par rapport aux réseaux de capteurs, les adaptations existantes pour réseaux de capteurs et nous terminerons par de nouvelles propositions de techniques de compression pour réseaux de capteurs : K-RLE et Aganmà³³.

Nous nous plaçons dans un contexte bien précis qui est celui d'applications non contraintes

³³Aganmà est un mot originaire du Bénin qui signifie le caméléon

par le temps pour la remontée d'informations, donc différente de la détection de feux de forêt par exemple qui est une application dirigée par événements où la contrainte temporelle est forte. Nous mettons l'accent sur des applications où les données collectées servent pour une analyse ultérieure comme dans le programme HYDRO-SENSOR-FLOWS dont le but est d'observer et quantifier les flux liquides et solides du glacier Loven Est (Spitsberg, 79°N).

Cette mise en contexte est très importante car il faut noter que le temps de compression constitue une activité donc un temps supplémentaire qui peut influencer sur les performances d'une application temps réel. Un autre paramètre important à noter est le fait que les plateformes avec lesquelles nous allons travailler, basées sur des microcontrôleurs TI MSP430, constituent les feuilles de notre architecture donc elles ont des capacités réduites, ce qui fait que les algorithmes classiques de compression ne sont pas applicables, il va donc falloir les adapter. Dans la prochaine section, nous allons tout d'abord présenter les principaux algorithmes de compression.

4.1 Les algorithmes de compression

Avec l'évolution des réseaux, la compression de données devient une technique de plus en plus répandue. Les algorithmes de compression se multiplient et deviennent de plus en plus performants. Le principal objectif d'un algorithme de compression est de réduire la quantité de données pour faciliter son stockage ou son transport. Dans le domaine des réseaux de capteurs, elle constitue une stratégie pour économiser l'énergie en réduisant le nombre de transmissions car c'est l'activité la plus coûteuse pour les capteurs [19, 122]. Aussi, en réduisant la quantité d'information, elle facilite son transport pour ces plateformes qui utilisent des technologies sans fil à faible débit.

Les deux principales opérations dans la compression de données sont : la compression et la décompression. Ces étapes permettent de définir deux catégories de compression : la compression sans pertes et la compression avec pertes. Lorsque l'objet avant compression est identique à celui après décompression, on parle d'un algorithme sans perte sinon lorsqu'il y a une différence entre les deux objets, on parle d'algorithme avec pertes. L'indicateur de performance d'un algorithme de compression s'appelle le taux de compression. La principale raison de l'utilisation d'algorithmes de compression avec pertes est le souci d'amélioration du taux de compression pour des données dont certaines informations peuvent être perdues sans affecter significativement le contenu, par exemple parce que les bits perdus ne représentent que du bruit.

4.1.1 Les algorithmes avec pertes

Les algorithmes de compression de données avec pertes ont pour principal objectif d'améliorer le taux de compression en éliminant quelques informations mais en restant le plus proche possible des données originales. Un exemple d'algorithme de compression avec pertes très répandu dédié aux images est JPEG. C'est un algorithme qui est disponible dans la littérature.

Cependant, comme le taux de perte est ajustable, plus la qualité de l'image est élevée, moins l'algorithme est efficace et inversement. Les images faisant partie des données les plus difficiles à transporter sur un réseau à cause de leur taille, des méthodes de compression permettant de réduire cette taille et donc de faciliter leur transport ont été mises en place. La principale contrainte est l'efficacité de la compression qui se fait au détriment de la qualité de l'image.

4.1.2 Les algorithmes sans pertes

Les algorithmes de compression de données sans pertes concernent les données dont l'usage doit conserver l'intégralité des informations et où toute modification peut avoir des influences néfastes sur les données définitives comme par exemple la modification d'un exécutable. Il existe trois principaux algorithmes de compression sans pertes définis dans la littérature et repris par différents formats de données :

- Huffman : L'algorithme de Huffman [123] est un codage statistiques dont le but est de calculer le nombre d'occurrences de chaque caractère de sorte que les caractères les plus fréquents soient codés avec le moins de bits. Cet algorithme se base sur une méthode de construction d'arbres binaires pour coder les octets.

- RLE : cet algorithme, acronyme de Run-Length Encoding, est une forme de codage de données très simple consistant à simplifier l'écriture de flux de données consécutifs qui se répètent (ex : la chaîne `aaaaaabbccccc` devient `7a2b5c` lorsqu'on applique RLE, ce qui constitue une chaîne plus courte).

Le principal inconvénient de RLE est son efficacité en terme de taux de compression qui dépend de la nature de données. En effet, les flux de données n'ayant pas de séquences répétées consécutives de valeurs donnent de mauvais résultats avec cet algorithme. Ex : Cannibalisation devient `1c1a2n1i1b1a1l1i1s1a1t1l1o1n`, ce qui constitue une chaîne plus longue. Cet algorithme est utilisé par les formats bitmaps : BMP, TIFF et PCX et permet de réduire les données graphiques redondantes.

- LZW : La méthode Lempel-Ziv Welch [83] est un algorithme très populaire, variant de LZ77, qui permet à l'aide d'un dictionnaire de données d'avoir des taux de compression intéressant. Contrairement à RLE, son efficacité n'est pas autant contraint par la source de données.

Cet algorithme reste incontestablement l'un des algorithmes les plus utilisés dans la compression de données en informatique. Aussi, contrairement à d'autres algorithmes de compression de données basés sur des dictionnaires, il va plus loin en ce sens qu'il n'est pas nécessaire au destinataire de disposer du dictionnaire de données pour décompresser l'information. Lempel-Ziv et ses variantes sont utilisés par plusieurs formats de compression tels que le standard de compression pour modem V24bis ou encore PKZIP.

L'utilisation d'un dictionnaire de données par LZW ou encore la construction d'arbres qui est la méthode utilisée par l'algorithme de Huffman ne sont pas adaptés pour les capteurs qui ont des unités de calculs limitées et des capacités mémoires réduites.

Cependant, il existe une adaptation du célèbre algorithme LZW pour RdCs nommé S-LZW (Sensor-LZW) [82]. Il faut noter que la compression de données dans les réseaux de capteurs sans fil reste encore un axe de recherche prometteur mais peu développé. La littérature sur la compression dans les réseaux de capteurs est pauvre et S-LZW est l'algorithme de référence dans le domaine.

4.2 L'algorithme S-LZW

L'adaptation de LZW pour réseaux de capteurs, nommée S-LZW (LZW for Sensor Nodes), a permis d'évaluer un algorithme de compression de données basé sur un dictionnaire pour les RdCs. Cependant, la construction d'un dictionnaire de données est un processus gourmand en mémoire, ce qui constitue de prime abord un obstacle à l'utilisation de ce type de techniques sur les capteurs qui ont des mémoires limitées.

De ce fait, pour adapter LZW aux capteurs, une nouvelle morphologie de cet algorithme s'articulant autour de trois principaux points corrélés a été défini. Ces points sont : la limitation de taille du dictionnaire, la limitation de la taille des données à compresser et la procédure à suivre lorsque le dictionnaire est rempli. Les principaux paramètres par défaut de S-LZW sont les suivants :

- Un bloc de 528 octets pour compresser les données et qui représente deux pages flash. En effet, S-LZW divise les flux de données d'entrée en blocs de taille fixe et compresse chaque bloc de manière indépendante.
- un dictionnaire limité à 512 entrées. Comme LZW, cet algorithme initialise le dictionnaire avec les 256 valeurs du code ASCII étendu, ce qui représente déjà 256 entrées. Pour chaque block utilisé dans la compression, le dictionnaire est re-initialisé. Une nouvelle chaîne dans le flux d'entrée crée une nouvelle entrée dans le dictionnaire, ce qui fait que le flux de données à compresser est limité. Cependant, différentes stratégies ont été développées afin de résoudre le problème de dictionnaire rempli. Deux options existent qui sont : soit le gel du dictionnaire et l'utilisation telle quel pour compresser le reste des données dans le bloc, ou il peut être remis à zéro et recommencer à zéro. Toutefois, ce problème ne se produit pas lorsque le flux de données à compresser est petit, et donc le dictionnaire n'est pas plein.
- Un mini-cache, qui est une table de hachage à 32 entrées est ajouté à S-LZW afin d'obtenir un avantage de la répétition des données du capteur. En effet, les capteurs ayant tendance à avoir des données répétitives sur des intervalles courts, le mini-cache associé au dictionnaire conserve les entrées du dictionnaire récemment utilisées et créées. L'index de hachage correspond aux quatre derniers bits de l'entrée du dictionnaire.

Malgré les adaptations qui précèdent, cet algorithme, basé sur un dictionnaire de données, a besoin d'une mémoire RAM significative et donc n'est pas utilisable sur toutes les plateformes telles que celles par exemple s'articulant autour d'un microcontrôleur TI MSP430F149 avec 2 Ko de mémoire RAM. Dans [82], les performances de S-LZW ont été évaluées sur un microcontrôleur TI MSP430F1611 avec 10 Ko de mémoire RAM.

C'est pour cette raison que nous allons définir de nouveaux algorithmes de compression qui sont utilisables par plusieurs types de plateformes quelque soit la taille de leur mémoire RAM. La question est donc de savoir : comment définir un nouvel algorithme de compression de données au moins aussi efficace que S-LZW et utilisant très peu de mémoire ?

4.3 L'algorithme RLE (Run-Length Encoding)

Run-Length Encoding (RLE) est un algorithme de compression simple et sans pertes qui a été très utilisé pour les images et l'envoi de fax. Comme décrit dans [124], l'idée de base de cet algorithme est la suivante :

Définition : Si un item d apparaît n fois consécutivement dans le flux de données d'entrée, nous remplaçons les n occurrences avec la simple paire nd .

La simplicité de RLE est un atout par rapport aux contraintes liées aux ressources des capteurs telles que la capacité mémoire limitée ou encore la puissance de calcul réduite. Cependant un problème perdure, celui de l'efficacité de cet algorithme. En effet, l'efficacité de cette technique de compression est étroitement liée à la nature répétitive des données d'où la question de savoir comment améliorer les performances de cet algorithme en conservant sa simplicité pour des données de statistiques différentes ? Pour répondre à cette question, nous allons tout d'abord nous intéresser aux variantes de RLE, même si, celles-ci sont très spécifiques aux images.

En effet, pour des images matricielles par exemple, il existe des variantes pour encoder de manière séquentielle soit colonne par colonne, en verticale, par flot 4x4 pixels ou encore en zigzag (Fig. 4.1). Cependant, toutes ces variantes sont spécifiques aux images matricielles et ce n'est que le type de parcours qui diffère.

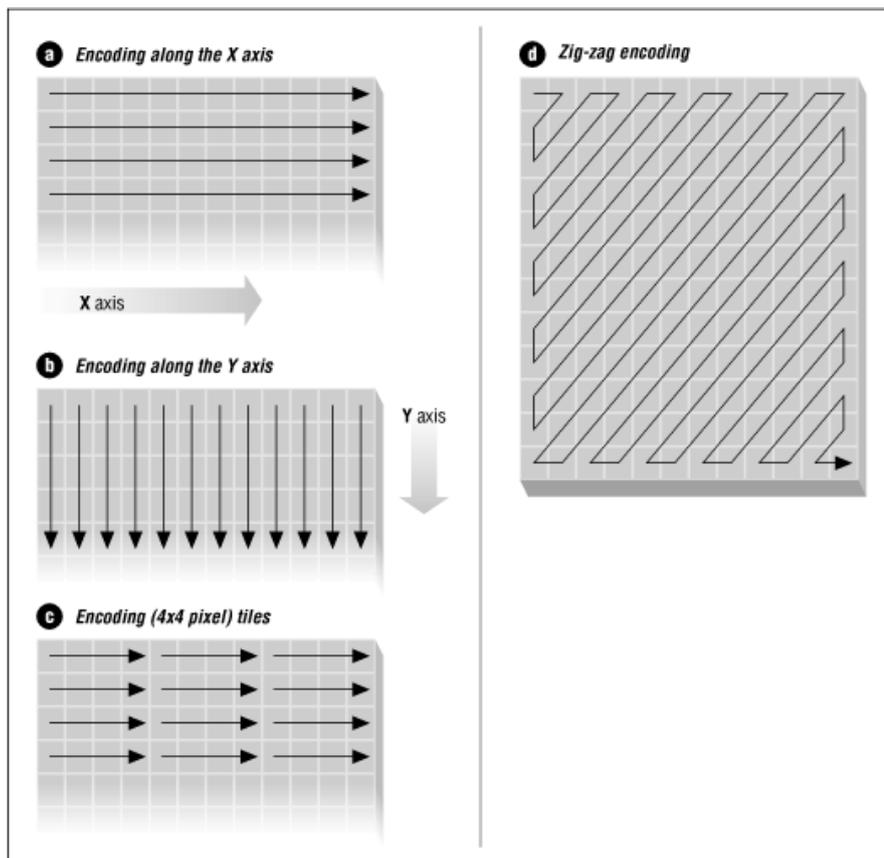


FIG. 4.1 – Les variantes de RLE [125]

Des variantes peuvent intervenir au niveau bits, octets ou pixels mais ne change pas l'algorithme en soit, c'est l'atomicité qui change.

Une autre variante de RLE est MNP5, utilisée par les modems et qui évite d'utiliser un caractère de compression spécifique. Le principe de base de cette variante est le suivant :

- un caractère pas répété (singleton) ou répété deux fois est codé tels quels,
- un caractère répété au moins trois fois est écrit 3 fois, puis suivi du nombre de ses occurrences restantes.

Exemple : bbbbbbvbbbvv donne bbb2vbbb0vv.

Cependant, les performances de MNP5 ne pourraient être supérieures à l'algorithme de référence dans le domaine des réseaux de capteurs qui est S-LZW. C'est pour cette raison que nous allons introduire un algorithme avec pertes pour accroître les performances de RLE quelque soient les statistiques des données.

Les variantes de RLE avec pertes sont très rares même s'il en existe qui ignorent des données par exemple en ne tenant pas compte d'un ou deux bits de poids faible pour chaque pixel. L'idée de cette variante qui est spécifique aux images constituera une piste intéressante dans nos choix.

Tandis que plusieurs variantes précédentes sont dédiées aux images et que l'efficacité de MNP5 dépend aussi de la nature répétitive des données, nous allons introduire un nouvel algorithme de compression de données qui conserve la simplicité de RLE (Fig.4.2) tout en améliorant ses performances quelque soient les statistiques des données. Cet algorithme, nommé K-RLE est donc inspiré de RLE et signifie RLE avec une précision de K.

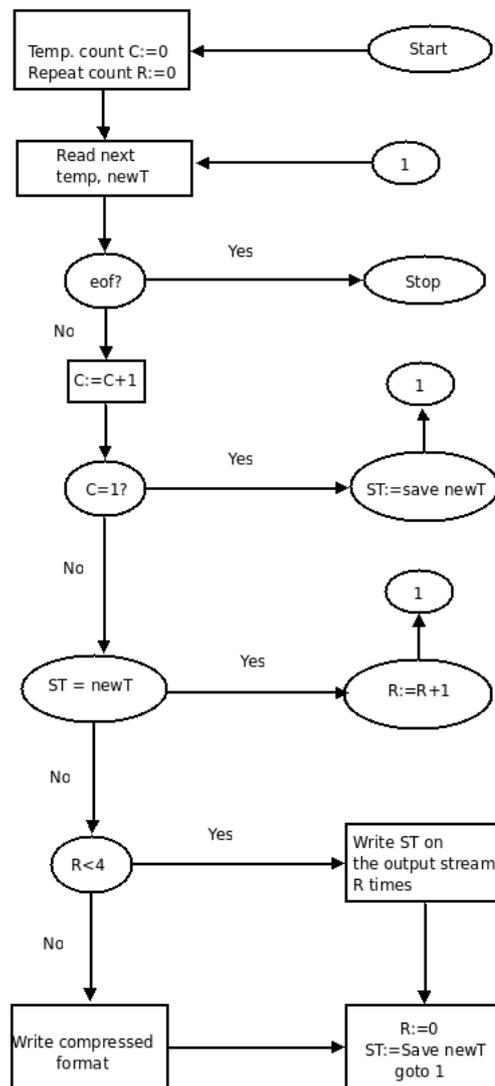


FIG. 4.2 – Algorithme de compression RLE [124]

4.4 Première contribution : l'algorithme K-RLE (K-Run-Length Encoding)

4.4.1 Principe

L'idée principale de l'algorithme K-RLE [126] est la suivante :

Définition : Soit K un nombre, Si un item d ou un item entre $d-K$ et $d+K$ apparaît n fois consécutivement dans le flux de données d'entrée, nous remplaçons les n occurrences avec la simple paire nd .

Dans cette définition, nous introduisons un nouveau paramètre K qui est une précision. Il faut noter que K a la même unité que le flux de données d'entrée, en l'occurrence des degrés pour

des températures. Lorsque K est proche de zéro, les résultats de K-RLE et RLE sont similaires.

Cependant, tandis que RLE est une technique conservative, c'est-à-dire sans pertes, l'introduction du paramètre K introduit une modification des données, ce qui fait de K-RLE une technique de compression non conservative, c'est-à dire avec pertes (Fig.4.3).

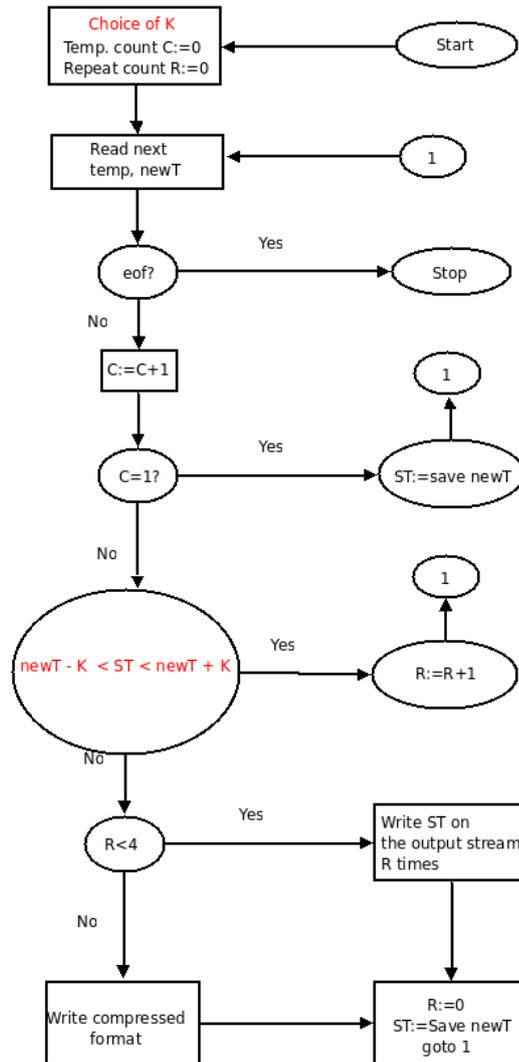


FIG. 4.3 – Algorithme de compression K-RLE

Néanmoins, il est important de constater que l'utilisateur peut définir K de sorte qu'une précision de K sur les données n'influe pas sur ses analyses, c'est-à-dire que les items d , $d+K$ ou $d-K$ soient considérés comme étant les mêmes.

Ainsi, nous allons introduire un nouveau concept *d'algorithme de compression sans pertes au niveau utilisateur*. Dans un premier temps, nous allons nous placer dans le cas bien précis de collecte de données scalaires par les feuilles de notre architecture (la température) et déterminer K de manière intuitive en considérant par exemple qu'un écart de 2° n'influencera pas considérablement sur les analyses d'un utilisateur final.

4.4.2 Résultats expérimentaux

Dans cette section, nous allons décrire les résultats obtenus en évaluant les algorithmes de compression précédents à l'aide d'un tableau de températures réelles de 500 octets. Ce tableau a été construit grâce aux données collectées sur le site Weather Underground [127] depuis le 1^{er} janvier 2008. Quatre différents lieux du globe ont été pris en compte : Libreville au Gabon, Cayenne en Guyane, Montbéliard en France et Svalbard en Norvège. Nous avons choisi ces différents lieux afin d'étudier le comportement des algorithmes précédents avec des variations de températures différentes (Fig. 4.4). Ces algorithmes seront évalués selon une représentation compatible avec une mesure réalisée par un microcontrôleur TI MSP430 dédié aux applications faible consommation.

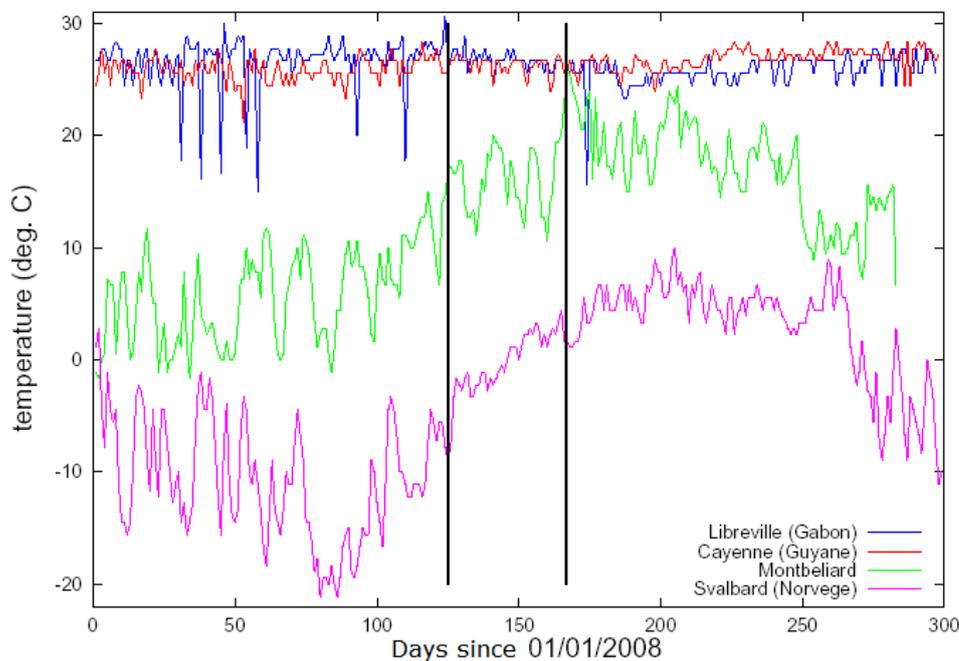


FIG. 4.4 – Représentation des variations de températures de différents endroits du globe

Comme décrit dans [128], le module ADC12 est présent sur les microcontrôleurs MSP430x14x et MSP430x16x. L'ADC12 est un convertisseur analogique-numérique 12 bit de haute performance dont les caractéristiques incluent un module de capture de températures. La fonction de transfert du capteur de température est la suivante :

$$VTEMP = 0.00355 * (TEMPC) + 0.986.$$

Grâce à cette fonction de transfert, nous avons simulé les séquences de mesures de températures en convertissant les archives quotidiennes de Weather Underground [127] en valeurs hexadécimales sur 12 bits telles que les auraient mesurées l'ADC de notre microcontrôleur TI MSP430. Ces températures sont donc codées sur trois octets :

b71 b7c b91 b7c b86 b71 b86 b86 b7c b86 ...

La Fig. 4.4 représente les archives de températures des quatre lieux allant de latitudes proches de l'équateur au cercle polaire arctique. Nous constatons que les variations de températures sont d'autant plus importantes qu'on s'éloigne de l'équateur. Nous allons donc évaluer les algorithmes dans ses différentes conditions réelles avec des variations de températures différentes.

Comparaison des taux de compression des algorithmes de compression

Dans cette partie, nous allons utiliser la formule du taux de compression en pourcentage pour estimer les performances des différents algorithmes de compression :

$$\tau(\%) = 100 * (1 - (\frac{taille_compressee}{taille_originale}))$$

Cette définition du taux de compression a l'avantage de donner 100% lorsque la compression est parfaite ou encore 0% lorsqu'il n'y a aucun gain c'est-à-dire que la taille compressée est la même que la taille originale.

Les contraintes mémoires du microcontrôleur MSP430F149 (2 ko de mémoire RAM) rendent impossible l'évaluation de l'algorithme S-LZW, c'est pour cette raison que nous allons étudier les performances des différents algorithmes sur le même système s'articulant autour d'un microcontrôleur MSP430F1611 (10 ko de mémoire RAM) pour pouvoir les comparer avec différentes statistiques des données.

Nous comparons dans un premier temps S-LZW et RLE (Fig.4.5). La Fig.4.5 montre que même si les variations des résultats de compression sont les mêmes, il y a une grande différence au niveau des taux de compression.

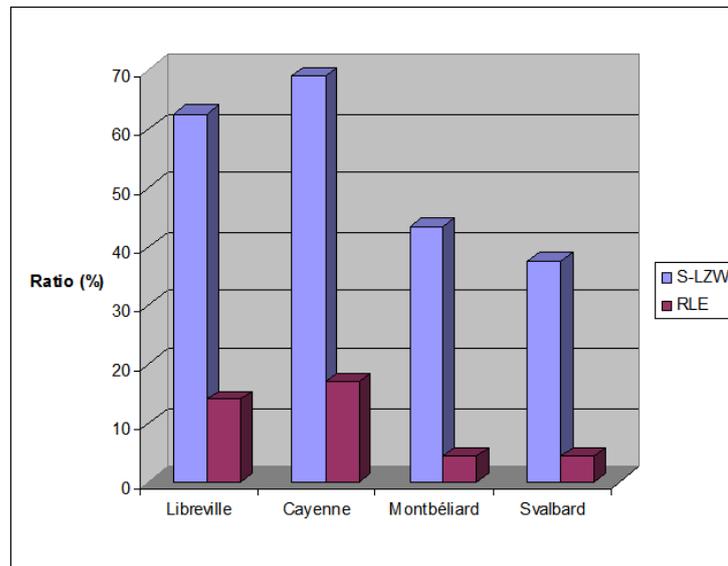


FIG. 4.5 – Comparaison entre S-LZW et RLE

En effet, pour chacun des algorithmes, les taux de compression sont de moins en moins bons lorsqu'on s'éloigne de l'équateur. Cependant, S-LZW offre de bien meilleurs résultats que RLE,

ce qui était prévisible mais intéressant à évaluer. Tandis que le meilleur taux de compression de RLE pour les quatre villes confondues est 17%, le taux moyen pour S-LZW est d'environ 53%.

Les résultats qui précèdent illustrent bien l'intérêt d'introduire un nouvel algorithme de compression simple et non gourmand en mémoire, capable non seulement d'être utilisé par plusieurs plateformes telles que celles basées sur un microcontrôleur TI MSP423F149, mais également capable d'être aussi performant que les algorithmes qui utilisent un dictionnaire de données et donc qui nécessitent de la mémoire comme S-LZW. Ces différents critères nous ont guidés vers une variante de RLE appelé K-RLE.

Rappelons que cet algorithme introduit un nouveau paramètre qui est K. Nous allons donc dans un premier temps définir un cas pratique où l'utilisateur considère qu'une différence de 2° entre les différentes températures n'est pas critique ($K = 2^\circ$). Lorsque $K = 0^\circ$, il n'y a aucune différence entre RLE et K-RLE. K-RLE devient tout simplement RLE, c'est pour cette raison que nous allons aussi choisir un K proche de zéro, soit $K = 0,5^\circ$ et les comparer par rapport à RLE.

La Fig.4.6 montre que 2-RLE fournit de meilleurs résultats que RLE. 2-RLE offre des taux de compression en moyenne 40% mieux que RLE. Cependant, la comparaison entre 1/2-RLE et RLE montre que les résultats des deux algorithmes sont les mêmes. De là, nous pouvons conclure en disant que tandis que K-RLE peut améliorer le taux de compression de RLE de manière considérable, le choix de K reste un facteur essentiel. Nous allons maintenant comparer les résultats de 2-RLE avec S-LZW.

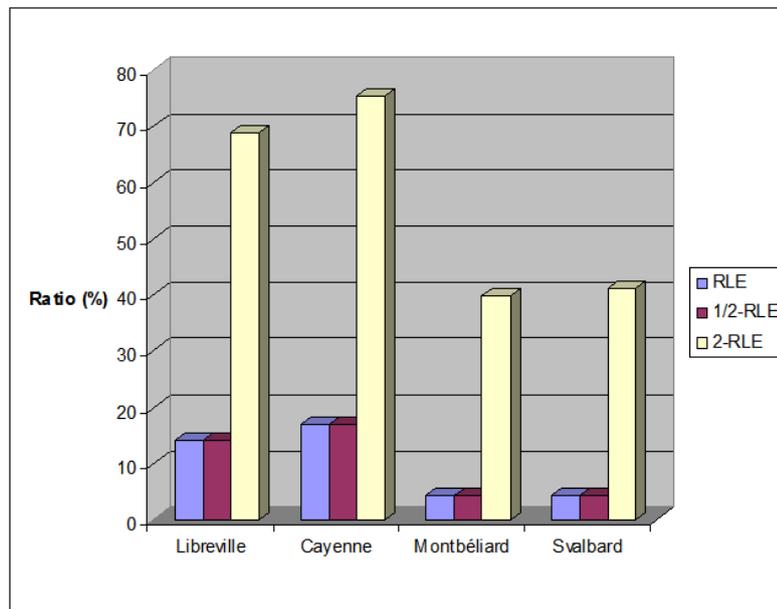


FIG. 4.6 – Comparaison entre RLE et K-RLE

La Fig.4.7 décrit les résultats de 2-RLE par rapport à S-LZW. Dans la majorité des cas, 2-RLE offre de meilleurs résultats que S-LZW. Le taux de compression moyen pour 2-RLE est de 56% pendant qu'il est de 53% pour S-LZW.

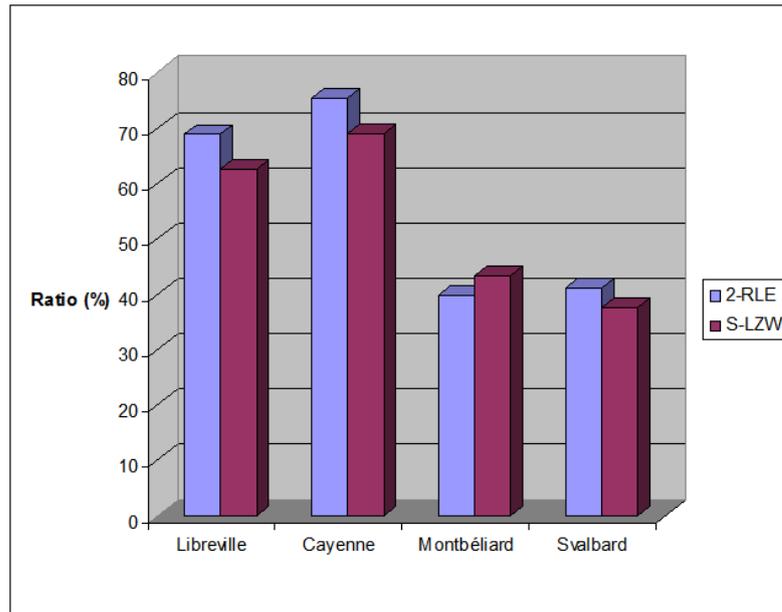


FIG. 4.7 – Comparaison entre S-LZW et K-RLE

Nous pouvons continuer à accroître le taux de compression de K-RLE en augmentant la valeur de K , mais il faut bien noter que l'efficacité de notre nouvel algorithme se fait au détriment de la qualité des données, car plus K est grand, plus la compression est efficace mais plus la différence entre les données originales et les données décompressées est importante, ce qui est une particularité des algorithmes de compression avec pertes. Cet algorithme se comporte comme JPEG qui est aussi ajustable et où l'efficacité de compression et la qualité des données évoluent de manière antagonistes.

Cependant, K-RLE ne perd pas les données en tant que tel mais plutôt les modifie avec une précision de K . Nous allons donc nous intéresser au taux de données modifiées pendant l'exécution de 2-RLE.

Le taux de données modifiées est le pourcentage de données rendues identiques à celles qui précèdent proches de K (Fig.4.8). Nous constatons que plus nous sommes proches de l'équateur, plus les variations de températures sont faibles, ce qui fait que le taux de données modifiées dans ces lieux est très élevé car les variations sont petites. Nous avons en moyenne 50% des données modifiées sur l'ensemble des zones étudiées, ce qui signifie que la moitié des données a été modifiée pour une précision de K égale à 2° .

Ces résultats montrent que le nouvel algorithme de compression K-RLE, inspiré de RLE, améliore les performances de RLE au détriment des données dont 50% en moyenne sont modifiées pour une précision de $K = 2^\circ$.

Cependant, contrairement à S-LZW qui est un algorithme de compression sans pertes et qui offre d'aussi bons taux de compression que 2-RLE, certes en moyenne 3% moins bons, K-RLE a beaucoup moins de contrainte mémoire et est utilisable sur plusieurs plateformes telles que celles basées sur un microcontrôleur TI MSP423F149 qui n'a que 2 Ko de mémoire RAM.

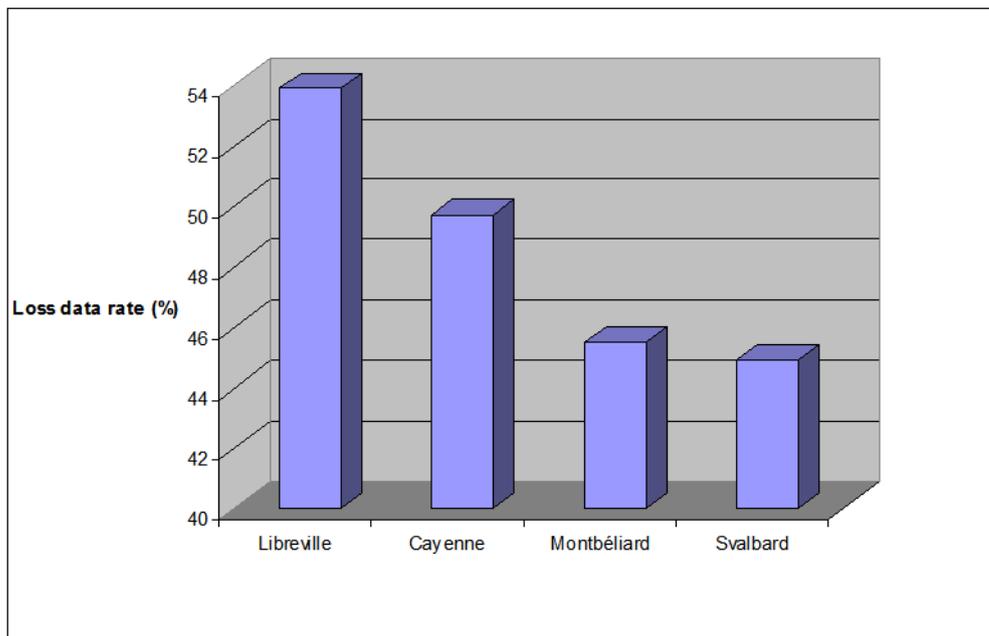


FIG. 4.8 – Représentation du taux de données modifié par 2-RLE

Après avoir évalué l'efficacité de nos algorithmes en terme de compression, nous allons nous intéresser à l'évaluation de la consommation énergétique pendant la compression.

Comparaison de la consommation énergétique des algorithmes de compression

Dans cette section, nous allons évaluer la consommation énergétique des algorithmes de compression précédents à l'aide d'un simulateur détaillé de plateforme matérielle appelé WSim et développé par une équipe du Laboratoire CITI, INRIA Rhône Alpes. Ce simulateur constitue une partie d'un environnement intégré pour le développement et le prototypage rapide des applications de réseaux de capteurs sans fil connue sous le nom de Worldsens [129].

L'intérêt de WSim est qu'il permet d'utiliser un code binaire classique comme si nous étions sur une réelle plateforme et facilite le debugage d'applications, ce qui fait de lui un émulateur de plateforme matérielle de réseaux de capteurs. En effet, nous avons utilisé les mêmes programmes développés sur notre plateforme matérielle sur WSim où un microcontrôleur MSP430F1611 est défini par défaut. Cet atout de WSim fait de lui un outil très intéressant pour débbugger.

La formule permettant d'estimer la consommation énergétique E en joules d'une activité à partir d'une puissance P (en Watt) et du temps T d'exécution de l'activité en secondes est :

$$E = P * T,$$

La puissance P (en Watt) est déterminée à partir de la tension U (en Volt) aux bornes du dispositif et du courant I (en Ampère) consommée pendant l'activité, soit :

$$P = U * I,$$

d'où

$$E = U * I * T,$$

Puisque nous connaissons la tension aux bornes de notre plateforme, qui est de 3,3 V, nous allons calculer le courant consommé à partir des différents modes de fonctionnement utilisés par le microcontrôleur TI MSP430 [128] et mesurer le temps d'exécution de chaque activité. Ces mesures vont être possible grâce au simulateur WSim qui définit les différents modes de fonctionnement du microcontrôleur et le temps d'exécution de chaque activité.

Après avoir défini la procédure d'évaluation de la consommation énergétique à l'aide du simulateur WSim, nous allons présenter les résultats obtenus pour les différents algorithmes dans les différentes zones.

La Fig.4.9 détaille la consommation énergétique des différents algorithmes pendant la phase de compression. Nous constatons que l'algorithme S-LZW étant sans perte avec des taux de compression intéressants, consomme le plus d'énergie. S-LZW utilise en moyenne 0,0224 mJ tandis que RLE consomme globalement 0,0053 mJ et 2-RLE 0,0103 mJ. RLE consomme moins que les deux autres algorithmes. Ces résultats mettent en lumière le compromis existant entre l'efficacité de compression et la consommation énergétique.

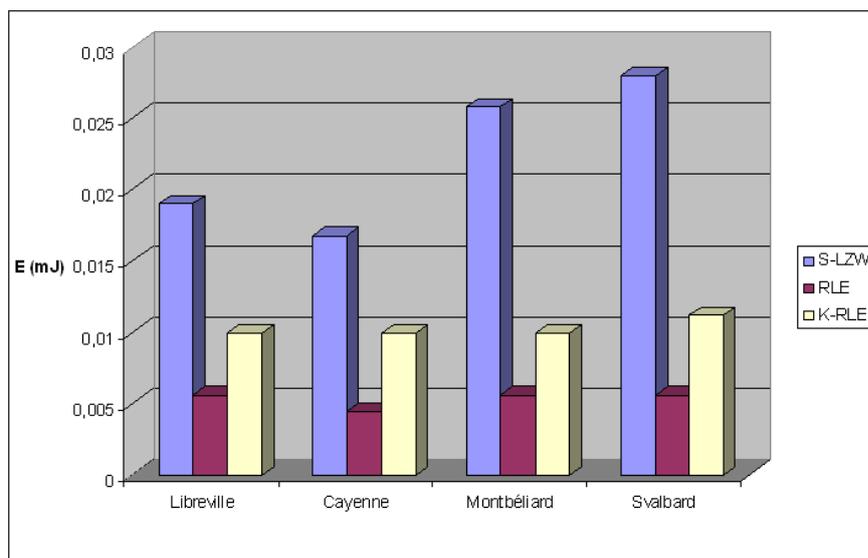


FIG. 4.9 – Evaluation de consommation de la phase de compression des algorithmes

Aussi, nous remarquons que pendant que RLE et 2-RLE ont des variations constantes au niveau de la consommation, S-LZW utilise plus d'énergie quand il y a beaucoup de variation au niveau des données d'entrée, typiquement pour des zones proches du cercle polaire arctique.

La Fig.4.10 montre que pendant que 2-RLE consomme plus que RLE pendant la phase de compression, il consomme beaucoup moins, environ 0,0011 mJ, pendant la phase de décompression. Cependant, S-LZW et RLE consomment respectivement en moyenne 0,015 mJ et 0,00165

mJ. La remarque précédente est encore confirmée, S-LZW consomme plus quand il y a beaucoup de variation au niveau des données d'entrée.

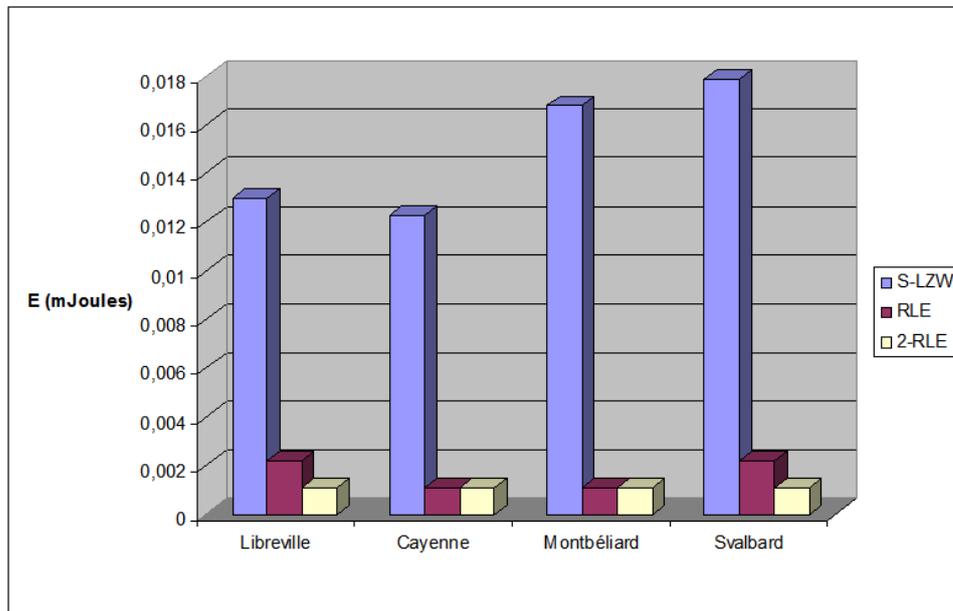


FIG. 4.10 – Evaluation de consommation de la phase de décompression des algorithmes

Récapitulatif

Ce travail nous a permis de comparer les résultats de l'algorithme de référence pour la compression de données dans les réseaux de capteurs, S-LZW, basé sur un dictionnaire de données avec l'algorithme RLE en utilisant différentes archives de températures de zones avec différentes variations.

Le fait d'avoir choisi des températures différentes est très important car dans [82], une comparaison entre S-LZW et RLE est faite mais avec des données d'entrée de RLE structurées pour améliorer ses performances d'où le nom que les auteurs lui donnent : RLE-ST, ST étant pour Structuré.

Aussi, la difficulté d'utilisation de S-LZW sur des plateformes avec des mémoires limitées, nous a poussé à introduire un nouvel algorithme de compression inspiré de RLE et nommé K-RLE. Cet algorithme offre de meilleures performances de compression par rapport à RLE et S-LZW.

Pour K égal à 2, cet algorithme de compression avec pertes offre des performances de compression 40% supérieure à RLE au détriment d'un pourcentage de données modifiée de 50%.

L'évaluation de la consommation montre que non seulement 2-RLE offre de meilleurs taux de compression que RLE et S-LZW, mais aussi, il consomme la moitié de la consommation de S-LZW qui utilise le plus d'énergie.

Ce travail met en valeur le compromis entre l'efficacité de l'algorithme de compression et la consommation énergétique. En effet, bien que S-LZW consomme le plus d'énergie, il est sans perte et il offre des taux de compression proches de ceux de 2-RLE qui sont les meilleurs. A contrario, RLE a les moins bonnes performances en terme de compression, mais il consomme moins.

Cependant, 2-RLE reste une évaluation de K-RLE pour un K précis qui a été fixé à 2 arbitrairement et appliqué à tous les différents endroits du globe de la même manière. Dans la prochaine section, nous allons tenter de caractériser la recherche d'un K optimal qui sera dépendant du lieu où K-RLE est appliqué.

Recherche d'un K optimal pour l'algorithme K-RLE

Dans cette partie, nous allons évaluer deux méthodes différentes permettant de caractériser K car les performances de notre nouvel algorithme K-RLE sont étroitement liées à ce paramètre. Nous allons nous focaliser sur l'écart-type et la variance d'Allan.

Ecart-type

Par définition, l'écart-type est la moyenne quadratique des écarts à la moyenne \bar{x} . Elle mesure, en statistiques et en probabilités, la dispersion d'une série de valeurs autour de leur moyenne et se définit comme suit :

$S = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$ avec n la taille de l'échantillon des valeurs $(x_1, x_2, \dots, x_i, \dots, x_n)$ où les x_i suivent la même loi que x, et \bar{x} représente la moyenne arithmétique $\bar{x} = \frac{1}{n} \sum_{i=1}^n (x_i)$.

Un autre indicateur lié à l'écart-type est la variance. La variance est le carré de l'écart-type et se note S^2 (ou σ^2). Elle se définit par conséquent comme la moyenne arithmétique des carrés des écarts à la moyenne \bar{x} . Il est donc facile de trouver la variance à partir de l'écart-type et vice versa.

La variance d'Allan

La variance d'Allan [130] est généralement utilisée pour caractériser la dépendance temporelle (nombre d'échantillons) d'une grandeur. Contrairement à la variance classique, c'est une variance terme à terme qui sert à estimer la stabilité dans le temps. Elle a été introduite historiquement pour caractériser l'évolution temporelle de la fréquence des oscillateurs.

Pour une suite d'échantillons $\{x_i\}_{0 \leq i \leq n}$, le variance d'Allan se définit comme suit :

$$\sigma_y^2 = \frac{1}{n-1} \sum_{k=1}^n [\bar{y}_k - \langle y_n \rangle]^2,$$

avec $\bar{y}_k(\tau) = \frac{1}{\tau} \int_{kt}^{(k+1)t} x(t) dt$

Il s'agit donc de calculer la variances des moyennes prises sur des billes d'échantillons τ variable. Il faut noter que la variance d'Allan est appliquée à différentes échelle de temps comme illustré sur la Fig. 4.11 et nous nous intéresserons au minimum estimé de la variance d'Allan.

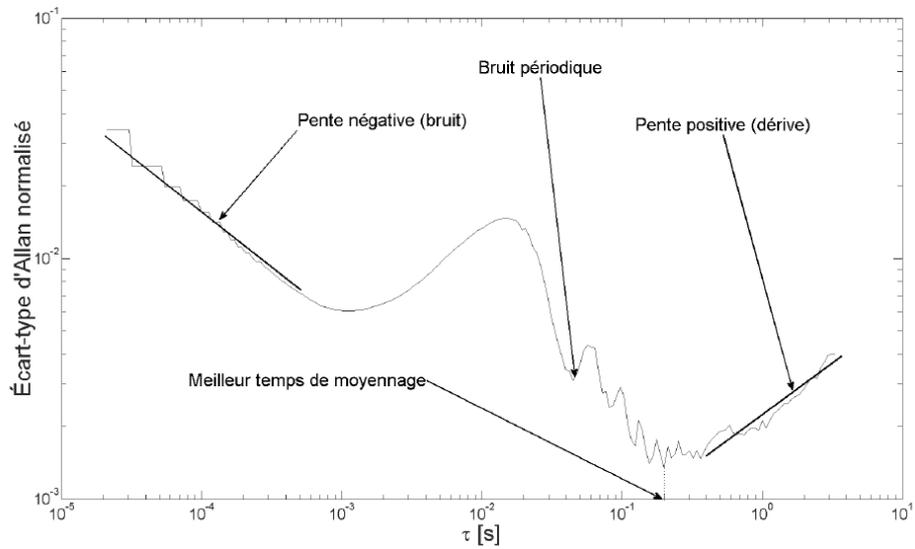


FIG. 4.11 – Courbe typique d'écart-type d'Allan [131]

Après avoir défini un nouvel algorithme de compression de données et différentes méthodes qui y sont associées, nous allons nous intéresser à l'évaluation des performances de cette nouvelle technique en la comparant avec les algorithmes existants.

La Fig.4.12 illustre la recherche des paramètres résultants de la variance d'Allan pour les différents lieux géographiques choisis précédemment. Nous constatons que plus les lieux sont proches de l'équateur, c'est-à-dire que les températures sont quasi-constantes et plus les résultats de la variance d'Allan sont proches de zéro.

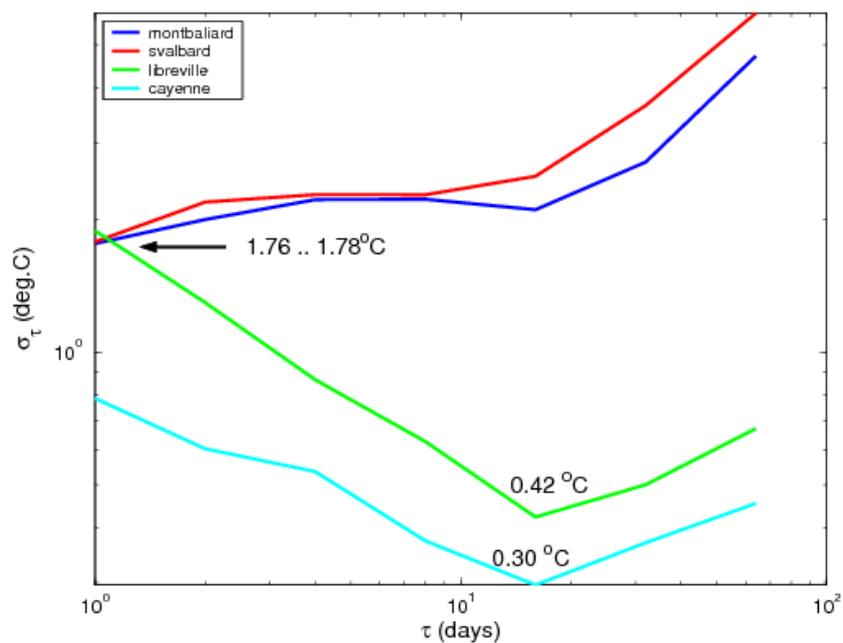


FIG. 4.12 – Variance d'Allan appliquée à différentes archives de températures

Le tableau récapitulatif 4.13 nous donne les résultats correspondants à chacune des mesures. Nous pouvons considérer qu'avant chaque déploiement l'utilisateur aura choisi K en fonction des variations de température de l'endroit où il souhaite déployer son réseau même si il faut reconnaître que l'idéal serait un K adaptatif en fonction des évènements.

Fonction / Archives	Libreville (Gabon)	Cayenne (Guyanne)	Montbéliard (France)	Svalbard (Norvège)
Ecart-type	1,9	1,1	6,8	7,9
Variance d'Allan	0,4	0,3	1,7	1,7

FIG. 4.13 – Tableau récapitulatif des résultats obtenus pour chaque fonction

La Fig.4.14 montre les performances de K-RLE respectivement en les comparant avec RLE et S-LZW. Les taux de compression de K-RLE en utilisant l'écart-type sont les meilleurs et atteignent jusqu'à 80% tandis qu'avec la variance d'Allan, les résultats sont en moyenne de 28 %. Pour les zones proches de l'équateur, c'est-à-dire où les températures sont constantes, la variance d'Allan donne des résultats proches de zéro ce qui fait que les performances de K-RLE sont les mêmes que RLE. Par contre, dans les zones éloignées de l'équateur et qui sont proches des cercles polaires avec beaucoup de variations de température, les résultats de K-RLE obtenus avec la variance d'Allan sont proches de ceux de S-LZW.

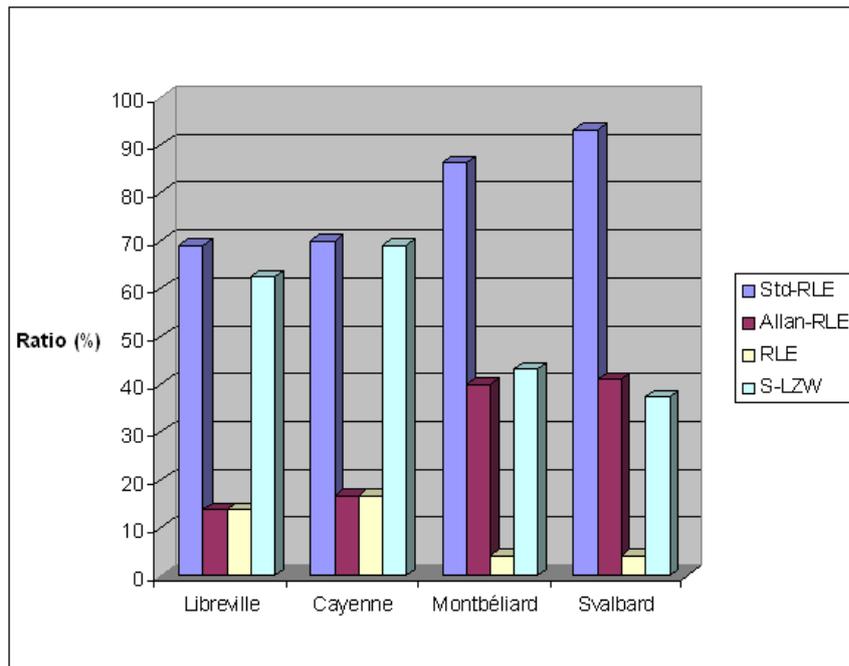


FIG. 4.14 – Comparaison entre S-LZW, RLE et K-RLE avec différentes valeurs du paramètre K

Un constat très intéressant est le fait que les performances de K-RLE en utilisant la variance d'Allan ou l'écarttype sont d'autant meilleures qu'on s'éloigne de l'équateur, c'est à dire lorsque les variations deviennent de plus en plus importantes. Ce constat diffère des algorithmes RLE et S-LZW dont les performances deviennent de plus en plus mauvaises quand on s'éloigne de l'équateur. Le point commun entre S-LZW et RLE qui peut expliquer ce phénomène est le fait

que ces deux algorithmes sont sans pertes contrairement à K-RLE. Pour continuer, nous allons observer les variations du taux de données modifiées par K-RLE (Fig.4.15).

La Fig.4.15 montre que tandis que K-RLE a des taux de compression satisfaisant avec les différents paramètres précédents, il a également des taux de données modifiées en moyenne d'environ 22 % et 71 % respectivement avec la variance d'Allan et l'écart-type. Ces taux deviennent d'autant plus grands que les variations sont importantes, c'est à dire lorsqu'on se rapproche des cercles polaires, allant jusqu'à 93 % de données modifiées pour Svalbard avec les valeurs de K fourni par l'écart-type. Nous constatons que la variance d'Allan bien qu'elle offre de moins bonnes performances que celle de l'écart-type, permet de conserver l'intégralité des données pour les zones où les températures sont quasi-constantes.

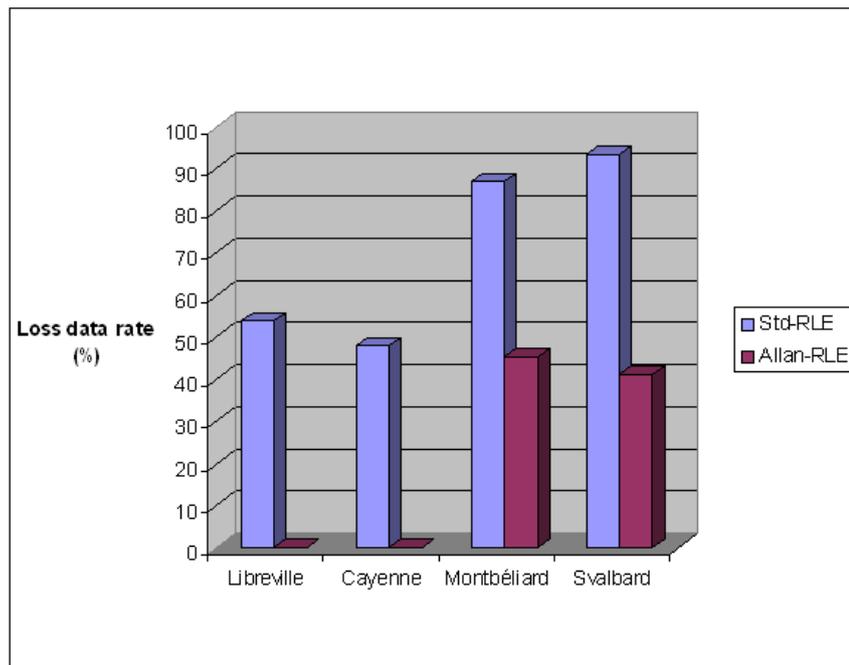


FIG. 4.15 – Comparaison des taux de données modifiées avec les paramètres de la variance d'Allan et de l'écartype (Std-RLE)

Récapitulatif

Le travail qui précède montre la pertinence du paramètre K dans l'utilisation du nouvel algorithme K-RLE. Lorsque nous utilisons les paramètres résultants de la variance d'Allan, nous constatons que nous pouvons conserver dans certains cas l'intégralité des informations au détriment des performances de compression et l'effet contraire est produit par les résultats de l'écart-type.

Cette étude a permis de caractériser le choix du paramètre K en mettant en lumière l'antagonisme entre l'efficacité de compression et la qualité des informations pour les algorithmes de compression de données. En effet, tandis que K-RLE est un algorithme simple, utilisable par plusieurs plateformes et permet d'améliorer de manière considérable les performances de compression, il le fait au détriment de données qui sont modifiées.

Énergie totale consommée (compression et transmission)

Après avoir défini et évalué un nouvel algorithme de compression de données, il est important de revenir au fil conducteur de l'utilisation de la compression de données dans les réseaux de capteurs qui est l'économie d'énergie. En effet, un des aspect très important de l'utilisation de la compression de données est la réduction de la consommation énergétique. C'est pour cette raison que nous allons évaluer l'énergie totale consommée par K-RLE dans les différents lieux avec les différentes valeurs de K résultant de l'écart-type et la variance d'Allan.

Pour évaluer l'énergie totale E_{Totale} consommée, il faut prendre en compte l'énergie de la compression $E_{compression}$ et l'énergie de la transmission $E_{Transmission}$ tel que :

$$E_{Totale} = E_{compression} + E_{Transmission}$$

A l'aide de WSim, nous avons donc mesuré le temps de compression et le temps de transmission des données compressée via une émulation du port série pour avoir un ordre de grandeur de la consommation totale. Nous allons décomposer l'équation précédente :

$$E_{Totale} = P_{microcontrôleur} * t_{compression} + P_{transceiver} * t_{transmission}$$

Il faut noter que $t_{transmission}$ est d'autant plus petit quand la compression est efficace et d'une manière générale $P_{transceiver} \gg P_{microcontrôleur}$. Puisque la puissance du microcontrôleur et celle du transceiver sont constantes, nous allons dans un premier temps nous intéresser aux variations des temps nécessaire pour la compression des données et leurs transmission.

La Fig.4.16 décrit les variations de temps pour chacune des villes en utilisant K-RLE avec d'une part des paramètres résultants de l'écart-type et d'autre part de la variance d'Allan.

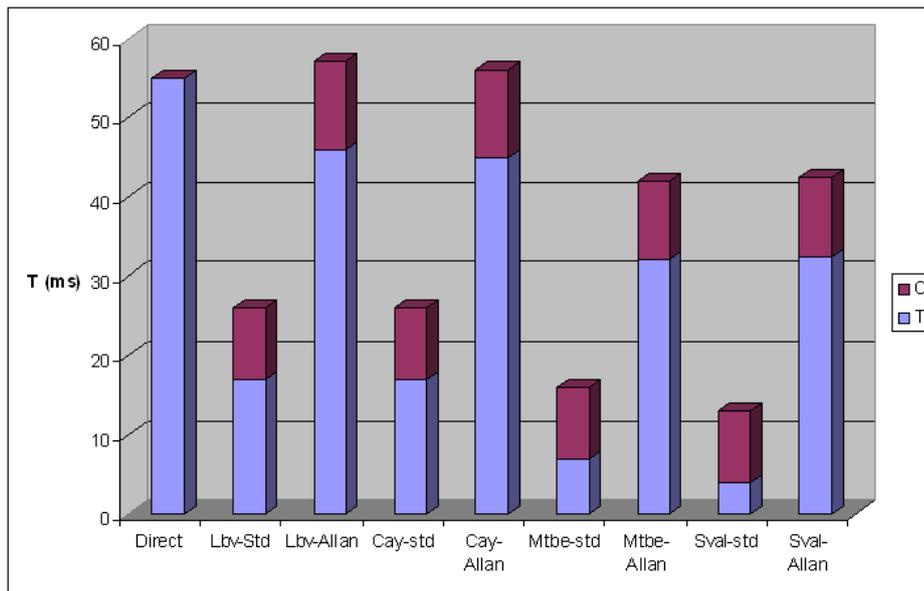


FIG. 4.16 – Evaluation des temps de compression et du temps de transmission des données compressées

Tandis que la variance d'Allan permet de maîtriser la modification des données, nous constatons que pour des villes comme Libreville et Cayenne proches de l'équateur, malgré le fait que le temps de transmission soit réduit par la compression, une transmission directe est plus judicieuse qu'une compression en terme de temps global.

Ces résultats montrent aussi qu'il n'est pas toujours intéressant de compresser. Néanmoins, dans les zones où il y a beaucoup de variation de températures, en utilisant la variance d'Allan, même si l'écart de temps entre une transmission directe et le fait de compresser et d'envoyer est minime, d'environ 10 ms en moyenne, l'avantage de compresser est quand même démontré. En effet, dans ce cas, la compression permet de gagner en temps de transfert.

Par contre, avec l'utilisation de K-RLE en prenant en compte l'écartype dans la majorité des cas nous avons en moyenne un gain de temps supérieur de moitié (50%) par rapport à une transmission directe. Nous constatons que le choix de K ne suffit pas, il faut prendre encore plusieurs autres critères en compte tels que l'endroit où l'algorithme sera appliqué et le fait de vouloir maîtriser ou non la modification des données. Les résultats qui précèdent montrent que dans certains cas, la compression de données peut aussi constituer un gain dans les applications contraintes par le délai.

Par la suite, nous allons évaluer la consommation énergétique en Joules, en prenant en compte les différentes puissances énoncées précédemment. Pour la puissance du microcontrôleur, nous allons nous appuyer sur les résultats précédents en considérant la consommation du MSP430 en mode actif à 1 MHz [128] :

$$\text{soit } P_{\text{microcontrôleur}} = U * I = 3,3V * 340\mu A = 1,12 \text{ mW}$$

Pour la puissance du module radio, nous utiliserons le tableau 4.17 défini dans [82], plus précisément sur le module le plus répandu qui est le Chipcon CC2420.

De là, nous avons $P_{\text{transceiver}} = 46,7 \text{ mW}$.

Module Radio	Range	Puissance de transmission	Puissance de réception
Chipcon CC2420	125 m	46,7 mW à 3 V	50,9 mW à 3 V
Chipcon CC1000	300 m	64,5 mW à 3 V	21 mW à 3 V
Chipcon MaxStream XTend	15 km	2,43 W à 5 V	444,5 mW à 5 V

FIG. 4.17 – Evaluation de la puissance des différents transceiver [82]

La Fig.4.18 donne l'évaluation de l'énergie totale (en mJoules) de l'utilisation de notre algorithme de compression de données. Ces résultats confirment le gain énergétique apporté par la compression de données dans les réseaux de capteurs car dans tous les cas de figures étudiés ici, compresser et envoyer est moins coûteux qu'un envoi direct.

Un autre constat intéressant est l'évaluation du poids de la transmission. En effet, ces exemples montrent clairement qu'une activité en local effectuée par le microcontrôleur est insignifiante par rapport à la transmission de données. Cela était prévisible rien qu'en comparant

l'écart considérable entre la puissance du transceiver ($P_{transceiver} = 46,7 \text{ mW}$) et la puissance du microcontrôleur $P_{microcontrôleur} = 1,12 \text{ mW}$.

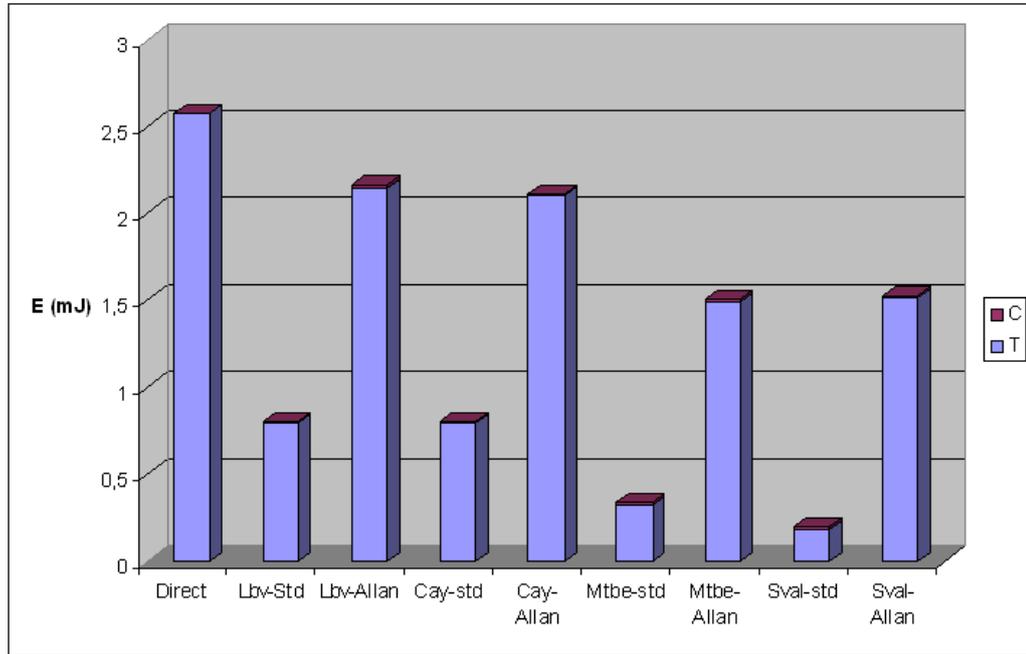


FIG. 4.18 – Evaluation de l'énergie totale de l'utilisation de l'algorithme de compression

4.5 Synthèse sur l'algorithme K-RLE

En résumé, cette étude nous a permis d'évaluer un nouvel algorithme de compression de données dans les RdCs efficace mais avec pertes. Cette évaluation illustre les caractéristiques de la compression de données dans les réseaux de capteurs et le gain de l'utilisation de cette technique de compression. Ce nouvel algorithme de compression de données inspiré de RLE et qui dépend d'un paramètre K est plus performant que l'algorithme de référence pour la compression de données dans les RdCs. Cependant, les performances de K-RLE se font au détriment des données car plus l'algorithme est efficace et plus les données décompressées s'éloignent des données originales. Ce résultat montre encore une fois qu'un moyen d'augmenter les performances d'un algorithme de compression passe nécessairement par la perte d'information.

Il faut aussi noter que non seulement cet algorithme est plus performant que l'algorithme de référence S-LZW mais surtout il consomme 50% moins en moyenne, ce qui fait de lui une nouvelle solution intéressante pour la compression de données dans les réseaux de capteurs.

Par contre, le choix de K dans la définition de notre algorithme reste encore un axe de recherche riche en perspectives. Dans ce travail, nous avons caractérisé K de deux manières en utilisant d'une part l'écart-type et d'autre part la variance d'Allan. Nous avons pu ainsi observer le comportement de notre algorithme avec différents paramètres.

Le tableau récapitulatif 4.19 décrit les caractéristiques de chacun des algorithmes étudiés. K-

RLE, qui est le plus performant d'entre eux, a un point faible qui reste la modification des données pour améliorer ses performances. Nous allons donc dans la prochaine section introduire un nouvel algorithme de compression simple et pas gourmand en mémoire capable de conserver l'intégralité des données contrairement à K-RLE : *Aganmà*.

Algorithmes	Avantages	Inconvénients
Sensor-Lempel Ziv Welch (S-LZW)	Sans pertes et efficace avec un bon taux de compression	Gourmand en mémoire
Run Length Encoding (RLE)	Très simple, rapide et sans pertes	Dépend de la nature des données
K-Run Length Encoding (K-RLE)	Très efficace, simple et rapide avec un bon taux de compression	Modification des données

FIG. 4.19 – Tableau récapitulatif des caractéristiques des algorithmes de compression précédents

4.6 Deuxième contribution : l'algorithme *Aganmà*

Dans cette partie, nous présentons un nouvel algorithme de compression de données avec les contraintes de départ fortes. En effet, nous voulons qu'il soit efficace, simple à utiliser, sans trop de contraintes mémoire et surtout qu'il assure la conservation des données.

Cette dernière contrainte a été ajoutée car elle constitue le principal point faible de K-RLE. Ce nouvel algorithme de compression s'appelle *Aganmà*, un mot originaire du Benin qui signifie caméléon.

La particularité d'un caméléon est son changement de couleur, nous avons donc choisi ce nom parce que notre algorithme va être adaptatif, c'est-à-dire qu'il va s'adapter en fonction des variations de température. Dans cette partie, la procédure d'évaluation de notre algorithme va être la même que la précédente, aussi nous n'allons insister que sur la définition de l'algorithme et passer directement à son évaluation puis à son interprétation.

4.6.1 Principe

Le principe de base de cet algorithme est d'optimiser la lecture séquentielle d'une liste de valeurs en codant les différences à l'aide de deux paramètres qui peuvent changer progressivement et régulièrement pendant la lecture des valeurs. Ces deux paramètres sont la valeur minimale appelée *min* et la valeur maximale appelée *max*.

La valeur *min* correspond à la première valeur lue, le but étant de retrancher à chaque valeur lue la valeur *min* et ainsi seulement la différence sera stockée. Lorsque la différence est négative alors nous avons rencontré un nouveau minimum et nous changeons ainsi la valeur de *min*.

Le fait de coder seulement les écarts constitue déjà une technique capable de réduire la taille de la chaîne de valeurs lues mais nous allons aller plus loin en ajoutant une contrainte supplémentaire qui est la fait qu'un écart ne doit être codé que sur 4 bits, ce qui signifie que sur un octet, c'est à dire 8 bits, nous allons avoir deux valeurs de température qui correspondront aux écarts de ces températures par rapport à un nombre référence *min* ou *max*.

Nous avons dans cette version de l'algorithme choisi arbitrairement 4 bits mais une optimisation de cet algorithme serait de choisir le nombre de bits pour coder en fonction des variations de températures, par exemple en prenant en compte la valeur de l'écart-type.

Une description de l'algorithme appliquée à une sequence de températures est la suivante :

Algorithm 1 Version simplifiée en pseudo-code de l'algorithme Aganmà

```
min ← lire(temp)
reference ← minimum
while non vide(tableauTemperature) do
  tempcourant ← lire(temp)
  if (reference == minimum) then
    ecart ← tempcourant − min
    if ecart < 0 then
      changeMin()
    end if
  else
    ecart ← max − tempcourant
    if (ecart < 0) then
      changeMax()
    end if
  end if
  if (ecart >= 0) && (ecart < 15) then
    Coder_Sur_4bit(ecart)
  else
    ChangeReference()
  end if
end while
```

Ce choix pour réduire la taille du bloc d'informations introduit deux contraintes supplémentaires :

1. sur 4 bits nous sommes limités à la valeur 15 comme écart maximal, c'est pour cette raison que nous allons introduire une valeur de référence *max*, qui lorsque l'écart sera supérieur à 15, va constituer la nouvelle valeur de référence et contrairement à *min*, où nous retranchons *min* à la valeur, nous allons retrancher la valeur lue à *max*. Si la différence par rapport à *max* est négative alors nous avons un nouveau *max*.
Lorsque l'écart avec la valeur *max* est une valeur supérieure à 15, nous utilisons la même technique et nous revenons à *min* et ainsi de suite. Cette caractéristique de l'algorithme où le *min* et le *max* peuvent changer régulièrement fait de lui un vrai caméléon.
2. le fait de jumeler en un seul octet deux écarts différents peut donner lieu à un nouveau caractère qui lui peut fausser la lecture pendant le décodage ou encore le codage d'un seul écart sur un octet dans la mesure où par exemple à la fin de la chaîne de valeurs il ne reste qu'un seul écart à coder sur 8 bits ou encore après avoir codé un écart sur les 4 premiers bits, l'écart suivant a une valeur supérieure à 15 ce qui veut dire qu'il faut changer de référence. Nous allons donc introduire un octet indicateur qui nous permettra de faire notre choix pendant le décodage.

Ce nouvel algorithme est très intéressant en ce sens qu'il est simple, il ne code que les écarts entre les températures et nous avons rajouté pour accroître ses performances le fait de diviser un octet en 2. Ce choix de 2 est le plus simple mais une perspective de ce travail serait justement de choisir la division de l'octet en fonction des variations.

4.6.2 Résultats expérimentaux

L'évaluation de ce nouvel algorithme va se faire dans les mêmes conditions expérimentales que celles qui précèdent en prenant comme sources de données les statistiques de températures des quatre villes présentées plus haut et en appliquant la fonction de transfert de l'ADC12 du MSP430 de sorte que les données lues soient codées comme si elles avaient été directement mesurées par un microcontrôleurs TI MSP430.

La Fig.4.20 compare les taux de compression d'Aganmà avec les algorithmes précédents.

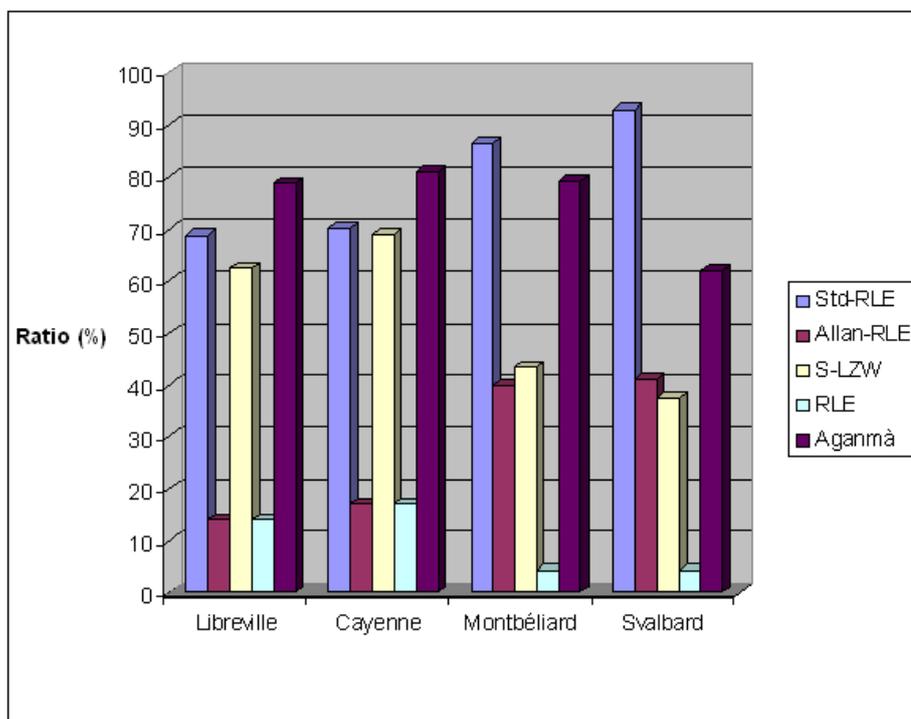


FIG. 4.20 – Comparaison entre Aganmà et les algorithmes précédents

Nous constatons que comme pour RLE et S-LZW qui sont des techniques de compression conservatives, les taux de compression d'Aganmà deviennent de plus en plus mauvais lorsqu'on s'éloigne de l'équateur.

Aganmà est non seulement un algorithme de compression sans pertes avec des meilleures performances que S-LZW, mais aussi il offre des résultats de compression aussi bons que Std-RLE, c'est-à-dire K-RLE en utilisant les valeurs du paramètre K fournies par l'écart-type. Cette nouvelle technique atteint en moyenne jusqu'à 75% quand l'algorithme Std-RLE atteint 80% mais avec des données modifiées.

Dans les zones proches de l'équateur où il n'y a pas beaucoup de variation de températures, nous constatons qu'Aganmà a les meilleures performances par rapport à tous les autres algorithmes confondus.

Par contre, lorsque les variations de températures s'amplifient, les taux de compression d'Aganmà se dégradent même si ils sont proches de ceux de Std-RLE qui deviennent meilleurs. Cependant, il faut noter que malgré cette dégradation, les performances d'Aganmà restent toujours au dessus de celles de S-LZW dans toutes les zones géographiques.

La Fig.4.21 donne l'évaluation énergétique des différents algorithmes pendant la phase de compression. Tandis qu'Aganmà a de bonnes performances de compression en conservant l'intégralité des données, il consomme plus d'énergie par rapport aux autres techniques pendant la compression et sa consommation évolue lorsqu'il y a de plus en plus de variations. Nous pouvons supposer que le fait que le caméléon change de couleur régulièrement est coûteux énergétiquement. En effet, dans les zones où il y a beaucoup de variations nous supposons qu'il y a beaucoup de changement de valeurs références *min* et *max*, ce qui consomme de l'énergie car l'activité devient plus longue.

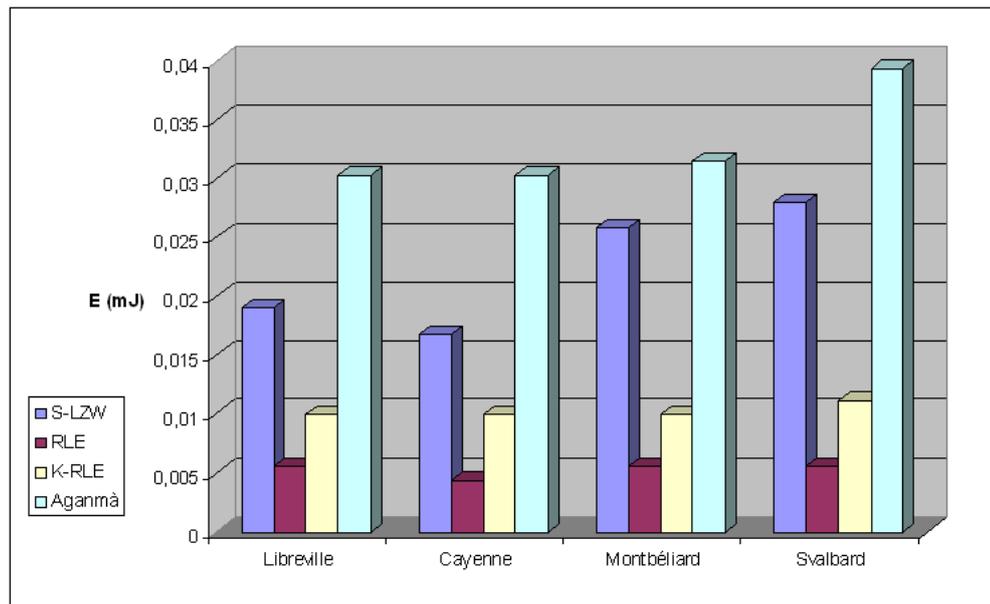


FIG. 4.21 – Evaluation de la consommation d'énergie pendant la compression

Comme décrit dans la démonstration de la section précédente, nous allons mettre l'accent sur l'énergie totale consommée mais dans un premier temps nous allons nous focaliser sur l'évaluation des temps de compression et des temps de transmission de l'information compressée.

La Fig.4.22 montre que la phase de compression avec Aganmà prend beaucoup plus de temps en général que la phase de transmission, ce qui fait que dans le cas où les variations sont nombreuses, il est préférable d'envoyer l'information directement. Std-RLE offre de meilleurs résultats en terme de temps.

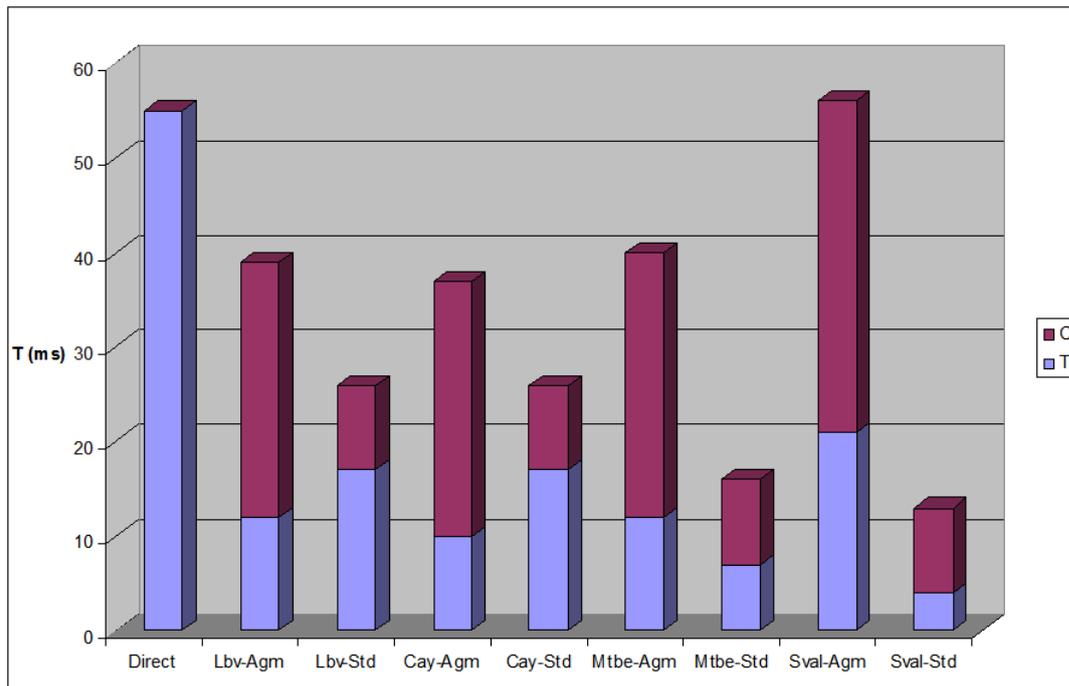


FIG. 4.22 – Evaluation du temps de compression et du temps de transmission avec Aganmà et Std-RLE

La Fig.4.23, qui est l'évaluation de l'énergie totale, montre que quelque soit l'algorithme de compression, la compression coûte beaucoup moins cher que la transmission. De ce fait, il est dans tous les cas intéressant de compresser les données avant de les envoyer sauf dans certains cas où il y a des contraintes de temps.

4.7 Synthèse sur l'algorithme Aganmà

En résumé, ce travail met en avant un nouvel algorithme de compression sans pertes, simple et efficace qui comble les lacunes de K-RLE. Ce nouvel algorithme de compression sans pertes est plus efficace que l'algorithme de référence sans pertes S-LZW.

Tandis que la compression avec Aganmà est couteuse énergétiquement par rapport aux autres, l'évaluation de l'énergie totale consommée montre qu'Aganmà de par son fort taux de compression dans les zones avec peu de variation consomme moins que Std-RLE et donc beaucoup moins qu'un envoi direct.

Ces performances énergétiques se dégradent dans les lieux où il y a beaucoup de variation par rapport à Std-RLE mais elles restent tout de même beaucoup plus intéressantes qu'un envoi direct avec un gain supérieur à 50%.

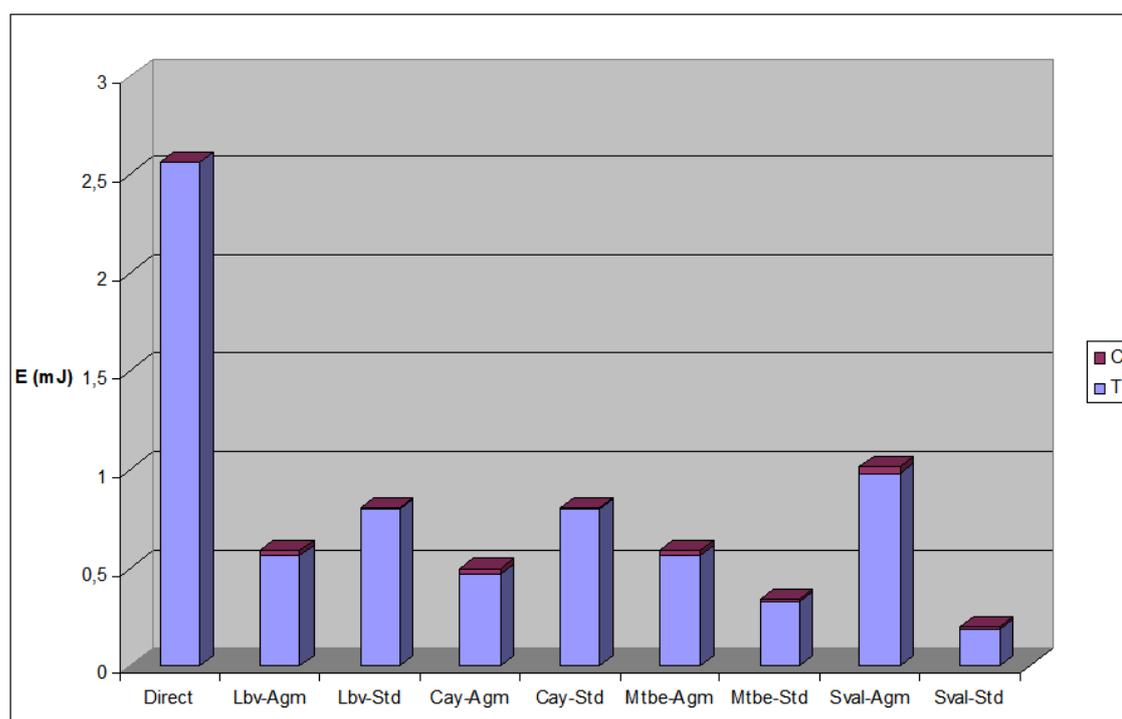


FIG. 4.23 – Evaluation de l'énergie totale d'Aganmà et Std-RLE

4.8 Conclusion

La compression de données dans les réseaux de capteurs se révèle être une technique efficace pour économiser l'énergie mais la mise en place d'un algorithme de compression performant dépend de critères souvent antagonistes. En effet, les ressources limitées des capteurs, les contraintes énergétiques, l'efficacité de compression et la conservation des données sont des critères souvent opposés.

Dans ce travail, nous avons proposé deux nouvelles techniques de compression de données : K-RLE et Aganmà. K-RLE est un algorithme de compression avec perte dont le niveau de perte dépend du paramètre K . Nous avons donc introduit différentes méthodes pour trouver K qui sont l'écart-type et la variance d'Allan. Tandis que l'application de K-RLE avec l'écart-type offre de bons taux de compression, la variance d'Allan permet de maîtriser la perte des données. Par contre, Aganmà est un algorithme de compression sans perte. Cette technique code les écarts sur 4 bits par rapport à une valeur de référence au lieu de coder les valeurs elles-mêmes, ce qui offre des taux de compression intéressants.

Toutefois, que ce soit pour K-RLE ou Aganmà, nous sommes capables d'avoir de meilleures performances par rapport à l'algorithme de référence S-LZW.

En somme, la compression de données, qui peut dans certains cas associée économie d'énergie et optimisation du temps de transfert des données par rapport à une transmission sans compression, ouvre des perspectives de recherche intéressantes dans les réseaux de capteurs.

Stockage de données dans les réseaux de capteurs

Sommaire

5.1	Présentation générale	121
5.2	Communication synchrone SPI et carte mémoire	122
5.2.1	Le Bus SPI	122
5.2.2	Support de TinyOS	122
5.2.3	Fonctionnement de la SD	123
5.2.4	Implémentation	123
5.3	Stockage en mémoire non volatile	124
5.3.1	Structure de la FAT16	125
5.3.2	Problématique	126
5.3.3	Implémentation	126
5.4	Description du module GPS	132
5.4.1	Structure des trames NMEA	132
5.4.2	Fonctionnement	132
5.4.3	Module GPS	132
5.5	Evaluation du module de stockage	133
5.6	Test et performances	134
5.6.1	Objectif	134
5.6.2	Conditions de l'expérience	134
5.6.3	Résultats expérimentaux	136
5.7	Conclusion	140

Le chapitre précédent a mis en valeur l'utilisation de la compression de données dans les réseaux de capteurs. Cette technique, qui est une expression de l'organisation du stockage d'information de manière temporaire sur support limité, a non seulement montré son gain énergétique en réduisant le nombre de transmissions mais aussi sa capacité à réduire de manière considérable le temps de transfert, ce qui permet de l'utiliser dans certaines conditions pour les applications contraintes par le délai. Cependant, il n'est pas toujours avantageux de compresser car un envoi direct peut dans certains cas prendre moins de temps que de compresser et envoyer. C'est un problème important à noter car le temps constitue un facteur déterminant pour certains types d'applications. Dans cette partie, nous nous focalisons sur des applications d'observation environnementale sans aucune contrainte temporelle et plus particulièrement dans un contexte semblable au projet HYDRO-SENSOR-FLOWS où les conditions environnementales rendent difficile la transmission d'information.

Dans ce contexte, la quantité d'informations à transférer est importante, en l'occurrence des images de haute qualité, et l'autonomie des dispositifs face aux contraintes de la transmission, c'est-à-dire le coût énergétique et les conditions environnementales, justifient le choix de supports de stockage de masse par rapport à une transmission. Cependant, ce choix nécessite une intervention humaine pour récupérer les informations contenues sur le support de stockage de masse. En effet, l'espace de stockage défini par défaut pour des plateformes de bas niveau étant limité, de l'ordre de la dizaine de kilo-octets, et la quantité d'information à stocker étant considérable, nous avons dû, avec l'aide de nos collègues électroniciens, ajouter sur notre plateforme expérimentale un dispositif de stockage de masse telles que des cartes SD³⁴.

L'évaluation de l'énergie totale a montré le poids énergétique de la transmission par rapport aux calculs locaux. Nous allons donc dans cette partie présenter un nouvel environnement de stockage d'informations développé sous TinyOS. Ce nouvel environnement permettra l'utilisation d'un support de stockage de masse en vue de récupérer les données sur n'importe quel système d'exploitation sans aucune intervention technique supplémentaire par l'utilisateur.

Le choix d'utilisation du stockage d'informations pour les réseaux de capteurs se justifie pour plusieurs raisons :

- le module de transmission est la principale source de consommation énergétique, ce qui pousse à se focaliser sur de nouvelles techniques capable de minimiser voire éviter la transmission d'information,
- la difficulté d'utiliser les technologies sans fil dans des conditions environnementales critiques comme celle du projet HYDRO-SENSOR-FLOWS pousse à chercher d'autres alternatives,
- le débit imposé par les technologies sans fil dédiées aux réseaux de capteurs qui consomment peu, multiplie considérablement le temps de transfert lorsque le phénomène observé nécessite une quantité importante de données à transférer,
- la robustesse du stockage.

Toutes ces raisons font du stockage d'information, une solution intéressante dans les réseaux de capteurs lorsque la disponibilité des informations n'est pas critique. Les bénéfices du stockage pour les réseaux de capteurs ont déjà été illustrés dans [84]. La particularité de notre nouvel environnement est sa compatibilité avec la grande majorité des systèmes d'exploitation modernes tels que MS Windows, Unix ou MAC OS X.

³⁴SD est un acronyme de Secure Digital

Ainsi, nous avons développé un driver de carte SD sous TinyOS-2.x en y intégrant le système de fichier FAT16 qui est peu gourmand en ressources et commun à plusieurs systèmes d'exploitation.

Le choix d'un tel système est motivé par plusieurs critères :

1. Il doit être économe en ressources. Ce qui exclu d'emblée tous les systèmes journalisés, qui bien que réduisant le risque de perte de données, entraînent l'augmentation du nombre de lectures, d'écritures et donc de traitements.
2. Il doit être multiplateforme afin que n'importe qui puisse nativement être en mesure de récupérer le contenu.

Nous avons donc choisi le système de fichiers FAT16 qui répond à nos exigences. Une implémentation de ce système de fichiers dédiée à la plateforme Shimmer [132] figure dans les contributions de TinyOS-1.x mais cet environnement est obsolète aujourd'hui. A l'heure actuelle, il n'existe aucune implémentation de ce système de fichiers sous TinyOS-2x, ce qui valorise notre choix.

5.1 Présentation générale

Dans le contexte du déploiement d'un grand nombre de capteurs pour surveiller les variations de quantités physiques, la radiofréquence est souvent la principale source de consommation énergétique et ainsi donc réduit le temps de vie du réseau. Ce temps de vie correspond au temps pendant lequel les capteurs sont capables de capturer des informations et de les remonter jusqu'à la station de base. La course à la gestion énergétique a entraîné le développement de technologies sans fil qui consomment peu mais qui ont des débits limités (de l'ordre d'une dizaine voire centaine de kilobits par seconde) pour un rayon de communication de dix à une centaine de mètres.

Certaines applications génèrent une quantité d'information conséquente supérieure à la dizaine de kilo-octets qui doit être stockée à chaque mesure mais pas nécessaire immédiatement telles que dans le projet HYFRO-SENSOR-FLOWS où les appareils photo numériques de haute résolution contrôlés par un microcontrôleur TI MSP430 stockent des images de haute qualité allant jusqu'à 10 MB par jour mais qui serviront pour une analyse ultérieure. Pour se conformer à l'application précédente, nous nous proposons d'utiliser des supports de stockage de masse bien répandus comme : MultiMediaCard (MMC) et Secure Digital (SD).

Pour faciliter l'analyse des données stockées sur ces supports pour les différents utilisateurs, nous avons choisi un système de fichier compatible avec la grande majorité des systèmes d'exploitation : FAT 16. Le système de fichier FAT existe depuis une trentaine d'année alors que les performances des ordinateurs étaient réduites avec des espaces de stockage de l'ordre d'une dizaine de kilo-octets comme celles qu'ont aujourd'hui nos capteurs, ce qui fait de ce système un outil idéal.

Nous allons donc décrire l'implémentation et l'évaluation du système de fichier FAT porté sous TinyOS-2.x. Ce travail va mettre en lumière un avantage de TinyOS qui est la portabilité. En l'occurrence ici nous avons évalué notre environnement avec des capteurs très répandus et fabriqués par Crossbow : TelosB et MicaZ. Pour ce faire, nous allons stocker une dizaine de Mega-octets de données provenant d'un GPS et simultanément d'autres informations telles que la température dans différents fichiers, ce qui va nous fournir une organisation structurée des

données compatibles avec les besoins des différents utilisateurs.

L'acquisition de coordonnées GPS est un bon outil d'évaluation des performances de notre environnement en ce sens que chaque mesure peut générer jusqu'à 1.5 MB par heure.

5.2 Communication synchrone SPI et carte mémoire

Les cartes SD et MMC dont les capacités évoluent très vite de nos jours vont constituer nos principaux supports de stockage. Nous allons donc, dans les lignes qui suivent, présenter les outils qui vont nous permettre d'accéder à ces supports.

5.2.1 Le Bus SPI

Le bus SPI (*Serial Peripheral Interface*) est une liaison série synchrone à 3 fils. Ces trois fils sont : une horloge générée par le maître, des données en sortie (du maître vers l'esclave) et des données en entrée (de l'esclave vers le maître). Cette liaison est dite synchrone car les différents périphériques du bus partagent la même horloge. De plus, un signal détermine quel périphérique du bus est actif : seul un périphérique peut être activé à un instant donné.

Ce protocole rapide, avec un débit de l'ordre de quelques Mb/s, est couramment disponible sur les microcontrôleurs et implémenté de façon matérielle dans différentes versions du MSP430, TI MSP430F149 ou TI MSP430F1611, en partageant des ressources avec un port de communication asynchrone (UART³⁵). Ainsi, l'utilisation du bus SPI nous prive du second port de communication asynchrone. L'implémentation matérielle du protocole SPI [128] ne fournit qu'un octet de mémoire tampon.

Contrairement au bus asynchrone RS232, les transactions sur le bus SPI sont initiées par le maître, qui est en général le microcontrôleur, et il ne peut donc pas y avoir de pertes de données en cas de délai dans la réponse à une transaction.

L'implémentation de la communication SPI pour le MSP430 ne sera pas nécessaire car elle est déjà disponible dans les routines d'exploitation du microcontrôleur. La seule tâche nécessaire étant de définir, à l'instar de l'UART, les configurations en terme de vitesse, d'horloge, etc... et la broche correspondant au signal d'activation du périphérique.

5.2.2 Support de TinyOS

Il n'existe pas à l'heure actuelle de driver TinyOS destiné à l'exploitation d'une carte SD. La seule implémentation existante était destinée à la version 1.x et cette solution ne répond pas au modèle hiérarchique de TinyOS-2.x. Il a donc fallu développer un nouveau driver parfaitement intégré et homogène dans la version actuelle du système, permettant sa totale indépendance vis à vis du matériel. Cette absence s'explique de par la structure des cartes nativement supportées par TinyOS : celles-ci n'offrent que du stockage non volatile grâce à des chipsets de mémoire.

³⁵UART est un acronyme de Universal Asynchronous Receiver Transmitter, c'est un composant matériel qui gère les transmissions via le port série.

5.2.3 Fonctionnement de la SD

Il existe deux manières de dialoguer avec la carte SD : soit à travers un protocole natif, soit à travers le protocole SPI. Les articles [133], [134] et le site [135] décrivent l'implémentation de protocoles de communication par SPI. Nous allons donc présenter notre implémentation spécifique d'un tel driver pour TinyOS-2x.

la carte SD offre les caractéristiques de lecture et d'écriture suivantes :

- La lecture d'un nombre d'octets allant de 1 à 512 a pour seule contrainte que l'adresse de début et l'adresse de fin soient contenues dans le même secteur. Le nombre d'octets par défaut étant 512, il est toutefois possible d'en fixer un au moment de l'initialisation ou lors de l'écriture du nombre.
- Écriture d'une taille fixée de 512 octets (1 secteur), l'adresse de début étant un multiple de 512.
- Effacement du contenu d'un nombre arbitraire de secteurs.

5.2.4 Implémentation

La première étape dans la conception du driver va consister à créer les fichiers qui feront la liaison entre le driver et le SPI du microcontrôleur. D'une manière générale, ces fichiers sont identiques à ceux utilisés par la liaison RS232.

La Fig. 5.1 présente le raccordement avec le module gérant le SPI sur le MSP430 (Msp430Spi0C). Le dialogue avec le matériel se fait grâce à l'interface SpiByte qui n'offre qu'une seule fonction `command uint8_t write (uint8_t tx)` qui envoie un octet et retourne la réponse du périphérique.

```

#include "hardware.h"
2 configuration PlatformSdC {
  provides {
4     interface SplitControl;
     interface SpiByte;
6  }
}
8 implementation {
  components projetSdP;
10  SplitControl = projetSdP.Control;

12  components new Msp430Spi0C() as SpiC;
  projetSdP.Msp430SpiConfigure <- SpiC.Msp430SpiConfigure;
14  projetSdP.Resource -> SpiC.Resource;
  //projetSdP.ResourceRequested -> SpiC.ResourceRequested;
16  SpiByte = SpiC;

18  components LedsC;
  projetSdP.Leds -> LedsC;
20 }

```

FIG. 5.1 – Description du module gérant le SPI sur le MSP430

La seconde étape consiste à définir de quelle manière sera faite l'interaction entre le driver SD et les couches supérieures. En effet, les contraintes mémoire d'un microcontrôleur impose une gestion particulière du système. Nous nous sommes donc basés sur un modèle classique de fonctions ne rendant la main qu'une fois l'action finie.

De là, le choix du fonctionnement pour les couches exploitant la SD sera laissée au soin du développeur. La Fig. 5.2 décrit l'interface d'utilisation du module SD. Les deux fonctions de cette interface prennent en paramètre l'offset en octet du début de la zone à écrire (ou à lire), la mémoire tampon contenant les données à écrire ou dans laquelle seront mises les données lues, et la taille lue.

```
2  /**
   * SdIO
   * sert à l'exploitation de la carte SD
   *
   */
6  interface SdIO {
8  /**
   * Commande pour la demande d'écriture d'une chaîne
10  * la commande est immédiate (au retour l'action est faite)
   *
12  * @param addr : adresse de début d'écriture
   * @param buf tableau à envoyer
14  *
   * @return SUCCESS Si la commande est acceptée
16  */
   command error_t write(uint32_t addr, uint8_t*buf);
18
20  /**
   * Commande pour la demande de lecture d'une chaîne
   * la commande est immédiate (au retour l'action est faite)
22  *
   * @param addr : position de début de lecture
24  * @param buf : tableau dans lequel mettre l'information
   * @param count longueur du tableau
26  *
   * @return SUCCESS Si la lecture est bonne
28  */
   command error_t read(uint32_t addr, uint8_t*buf, uint16_t *count);
30 }
```

FIG. 5.2 – Interface d'utilisation du module SD

Une fois le mode SPI validé, nous allons nous intéresser à la description de la FAT.

5.3 Stockage en mémoire non volatile

L'implémentation d'un système de fichiers sous TinyOS a pour principal objectif la compatibilité avec d'autres OS, ce qui facilitera l'analyse des données stockées par les différents utilisateurs. C'est pour cette raison que nous allons coupler l'utilisation d'une carte mémoire amovible à un système de fichiers, ce qui va nous procurer les avantages suivants :

1. Réduire le temps d'arrêt du capteur. Ainsi la carte peut être simplement échangée par une autre. Sinon, il serait nécessaire de passer par une communication RS232.
2. La carte sans système de fichier peut être utilisée en RawWrite, c'est-à-dire en faisant une écriture brute, mais cette solution impose un post-traitement avant son exploitation. Cela imposerait non seulement à l'utilisateur une certaine compétence mais aussi il faut noter que tous les systèmes d'exploitation n'autorisent l'accès à un support de stockage qu'à travers une partition.
3. Le système de fichier permet un transfert in situ, autorisant la libération de la carte et sa réutilisation immédiate.
4. Dans le cas d'un redémarrage sur une carte mémoire non vierge, le capteur ajoutera les informations à la fin du fichier existant.
5. L'utilisation de fichiers sur la partition permet une cohérence dans les données acquises. Par exemple, pour une application faisant l'acquisition de trames GPS et de températures, les deux informations ne se trouveront pas forcément dans le même fichier.

Tout support de stockage, pour être compatible et exploitable sur un ordinateur, doit comporter au tout début un Master Boot Record (MBR) qui fournit des informations relatives au support en lui même et les informations sur les partitions du support. Dans le cas de l'implémentation dans TinyOS, la seule information nécessaire se trouve être le numéro du premier secteur de la partition utilisée.

La formule pour trouver cette information est :

$\text{debPartition} = (*(uint32_t *)\&\text{buf}[446+8]+((\text{numPart}-1)*16))$ où 446 correspond à la position de la table de partition, 8 est le décalage pour obtenir le numéro du secteur de début de partition et 16 est la taille d'une entrée de partition.

5.3.1 Structure de la FAT16

Les détails de la FAT sont documentés sur [136] et [137]. Il ne sera donc fait mention que des points importants de sa mise en œuvre.

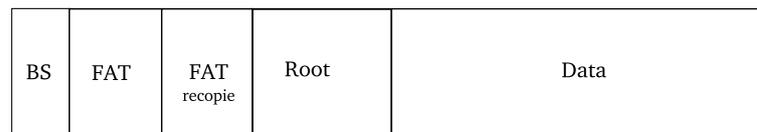


FIG. 5.3 – Structure d'une partition FAT

La Figure 5.3 présente la structure d'une partition FAT. Ces principaux composants sont :

- le BS (Boot Sector) qui contient l'ensemble des informations sur la partition,
- les deux tables d'allocation de fichiers qui contiennent les listes chaînées du contenu de chaque fichier,
- le répertoire racine (Root) qui contient une entrée par fichier,
- La zone des données (Data) qui contient les fragments des données du fichier à laquelle on accède grâce aux informations de la liste chaînée du fichier contenu dans la zone FAT.

A l'exception du Boot Sector, qui est l'équivalent du MBR pour une partition FAT, les autres zones ont une position et une taille définie dans le Boot Sector.

L'unité de la plus petite zone mémoire de la FAT est le *cluster*. C'est un agglomérat d'un certain nombre de secteurs définis dans le Boot Sector de la partition. Il est toutefois possible d'écrire des données plus petites que cette zone. Comme la FAT16 ne peut contenir qu'un maximum de 65525 clusters, plus la taille de la partition est grande, plus il est nécessaire que la taille d'un cluster soit importante.

La table d'allocation des fichiers est une zone contenant des listes chaînées de clusters. Ces listes chaînées correspondent à l'ensemble des fragments du fichier. Le cluster de début de la liste est défini dans l'entrée de fichier contenu dans le répertoire racine. La dernière zone avant les données est le répertoire racine. Celui-ci contient les entrées pour chaque fichier ou répertoire contenu dans la partition. Pour chaque fichier, il existe à la fois un ensemble de blocs définissant son nom long, ainsi qu'un bloc contenant à la fois son nom court et les informations tels que la taille, la date de création et le numéro du cluster de début de son contenu.

5.3.2 Problématique

L'implémentation d'un système de fichiers n'est pas quelque chose de trivial, surtout pour ce type d'environnement. D'une part, les caractéristiques du microcontrôleur imposent une économie en terme d'utilisation mémoire et d'autre part, la carte mémoire impose un traitement de l'information par paquet de 512 octets. A cela s'ajoute la vitesse du MSP430 qui réduit le débit des données.

L'utilisation d'un système de fichiers augmente le traitement par rapport à du RawWrite, il devient par exemple nécessaire de remettre à jour les structures de données relatives au fichier ou à la table d'allocation.

De prime abord, l'approche la plus attrayante, utilisée sur un ordinateur, consiste à ne pas mettre à jour systématiquement les informations. Le changement de la taille du fichier (2 octets) imposant la lecture/écriture de 1024 octets, soit pour les deux tables d'allocation, entraîne un ralentissement important de la vitesse d'écriture.

Toutefois cette solution, dans le cas d'un capteur, n'est pas du tout adaptée car elle entraîne une contrainte supplémentaire qui est l'augmentation de l'espace mémoire utilisée, qui ne sera donc pas disponible pour l'exécution en elle-même.

De plus, en cas de coupure de l'alimentation, l'état du système de fichier ne peut être garantie et un certain volume d'informations peut être perdu.

C'est pour cette raison que nous avons opté, au dépend de la vitesse et de la consommation liée à l'accès à la carte, de privilégier au maximum l'état de cohérence du système de fichiers.

5.3.3 Implémentation

Pour l'exploitation de la FAT trois drivers ont été développés. Tout d'abord en commençant par la couche la plus proche du matériel (Fig. 5.4) :

- MBR³⁶ concerne uniquement la partie spécifique au support, afin d'en extraire la position de début d'une partition. Cette information sera ensuite ajoutée à chaque demande de lecture ou écriture pour donner la position absolue du début du support.
- FAT concerne la partition en elle-même.
- File concerne le descripteur de fichier.

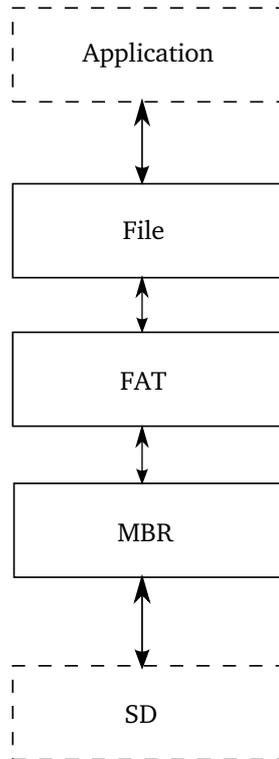


FIG. 5.4 – Schéma du stockage

Les spécifications les plus importantes concernent le module FAT en lui-même. En effet, pour être en mesure d'obtenir l'ensemble des informations, il est nécessaire d'exploiter toutes les données pertinentes contenues dans le *Boot Sector*.

La seconde opération importante concerne la réservation d'un cluster lors de l'utilisation d'un fichier vide ou quand le programme a complètement rempli le cluster.

L'algorithme est composé de deux parties essentielles :

Dans un premier temps, la recherche d'un cluster vide disponible s'effectue. Il est donc demandé tout d'abord une copie du secteur dans lequel se trouve le suivant du dernier cluster vide trouvé. Ensuite, la fonction va boucler tant qu'elle n'aura pas trouvé une case décrivant un cluster vide, avec au besoin un changement de secteur.

La seconde partie de l'algorithme concerne la mise à jour des deux tables d'allocation :

la première étape consiste à noter dans le tableau le cluster comme utilisé ($0xffff$), puis de rechercher la position du précédent cluster pour le fichier. Celui-ci contiendra le numéro du clus-

³⁶MBR est un acronyme de Master Boot Record et correspond au premier secteur adressable d'un disque. Il contient la table des partitions.

ter suivant, c'est à dire de celui qui vient d'être réservé.

A ce niveau, deux possibilités existent :

1. Les deux clusters sont dans le même secteur, dans ce cas l'écriture du secteur concerné modifiera l'ensemble.
2. Les deux clusters sont dans des secteurs différents, il faudra donc les mettre à jour.

Le dernier point important concerne la recherche de l'entrée d'un fichier dans le répertoire racine de la partition, il existe plusieurs cas :

- Le premier cas correspond à la fin de la liste des fichiers.
- Le second cas correspond aux noms de fichiers longs.
- Le dernier cas correspond à une entrée de fichier valide. Dans cette situation, le nom passé lors de l'instanciation de l'objet descripteur de fichier est comparé avec le nom trouvé. Si l'entrée est trouvée, alors la boucle s'arrête.

Une fois l'entrée trouvée, il faut en exploiter les renseignements. En effet, un fichier vide n'a pas encore de cluster attribué. Si c'est le cas, un cluster lui est automatiquement attribué, et l'entrée de fichier est remise à jour. Sinon, les informations sur la fin du fichier sont recherchées afin d'être en mesure de pouvoir directement exploiter le fichier lors de la première lecture.

Une fois la communication en mode SPI reconnue par la carte mémoire, le système de fichiers FAT présenté précédemment est activé. Une limitation de notre implémentation réside dans le fait que les fichiers ne peuvent pas être créés par le système de fichiers : des données peuvent être lues, écrites ou annexées à des fichiers existants, ce qui nous pousse à créer les fichiers à utiliser au préalable. La taille du fichier est mise à jour après chaque étape d'écriture.

Étant donné que la taille des blocs de base pour accéder à une carte SD ou MMC est de 512 octets, nous avons besoin à tout moment d'au moins deux mémoires tampons pour contenir les données manipulées par le fichier de stockage, typiquement, les prochaines données qui seront écrites sur le fichier ou encore une copie de la FAT qui sera mise à jour. Nos optimisations actuelles ont été validées sur les versions du microcontrôleur TI MSP430F149 (2 ko de RAM) et TIMSP430F1611 (10 ko de RAM), et sur un ATmega128 (4 ko de RAM).

Dans la partie qui suit, nous allons décrire les deux principaux programmes de notre implémentation. La Fig. 5.5 définit quels sont les fichiers utilisés et les périphériques nécessaires à l'accès à la carte SD tandis que le fichier associé décrivant l'application (Fig. 5.6) montre quelles sont les différentes couches logicielles auxquelles nous faisons appel pour pouvoir effectuer un stockage sur un support MMC ou SD. La FAT est tout d'abord initialisée, ouvre le fichier et ajoute les données (dans cet exemple une chaîne constante statique). Le fichier est clôt lorsque l'écriture des données s'est réalisée avec succès et se termine enfin la clôture du système de fichiers FAT (équivalent à démonter le système de fichiers sur les systèmes d'exploitation UNIX).

Les données sont immédiatement écrites sur la carte SD : même si la mémoire tampon a une taille inférieure à 512 octets, le dernier cluster contenant des données à partir de ce fichier est lu, les nouvelles données y sont donc ajoutées et la mémoire tampon est mise à jour et écrite sur la carte.

Cette stratégie minimise les risques de perte de données due à une défaillance du matériel (perte de puissance par exemple) et est compatible avec la plupart des applications de capteurs

embarqués dans lesquels différentes données collectées telles que des paramètres environnementaux n'atteignent pas une taille de 512 octets.

```

configuration fatTestAppC {}
2 implementation {
  components fatTestC as App, MainC, fatC;
4   components new TimerMilliC() as Timer0;
  components new fileC("toto.txt");
6   App.Boot -> MainC.Boot;
  App.Timer0 -> Timer0;
8   App.fatControl -> fatC;
  App.file -> fileC;
10  App.fileControl -> fileC.fileControl;
  fileC.fat -> fatC.fat;
12 }

```

FIG. 5.5 – Fichier de configuration

```

1 module fatTestC {
  uses {
3   interface Boot;
   interface Timer<TMilli> as Timer0;
5   interface SplitControl as fatControl;
   interface SplitControl as fileControl;
7   interface fat;
   interface file;
9   }
  }
11 implementation {
  event void fatControl.startDone(error_t err){
13   if (err == SUCCESS) call fileControl.start();
  }
15  event void fileControl.startDone(error_t err) {
   if (err == SUCCESS)
17   call file.write("hello world",11)
  }
19  event void file.writeDone(error_t err) {
   if (err == SUCCESS) call fileControl.stop();
21  }
  event void fileControl.stopDone(error_t err) {
23   if (err == SUCCESS) call fatControl.stop();
  }
25  event void fatControl.stopDone(error_t err){}
  event void Boot.booted() {
27   call Timer0.startOneShot(500);
  }
29  event void Timer0.fired() {
   call fatControl.start();
31  }
  }
}

```

FIG. 5.6 – Accès à la carte SD

Pour évaluer notre nouvel environnement de stockage de données, nous avons acquis des coordonnées GPS sur un intervalle court de l'ordre de la seconde. Que ce soit avec une fréquence de mesure par le GPS toutes les secondes ou toutes les minutes, la quantité de données générée est de l'ordre du Mega-octet.

Nous nous sommes donc proposés de travailler sur un microcontrôleur supportant TinyOS, précisément sur un circuit spécifiquement développé par nos collègues électroniciens pour une application permettant l'acquisition de données d'un récepteur GPS afin d'être stockées sur une carte mémoire utilisant un système de fichier compatible avec n'importe quel ordinateur.

La plateforme (Fig. 5.7) servant de base à nos évaluations, dispose des caractéristiques suivantes :

- Elle s'articule autour d'un microcontrôleur TI MSP430F149 disposant de 60 KB de mémoire non volatile FLASH et de 2KB de RAM.
- Elle est équipée d'un récepteur GPS ET312 et de son antenne.
- Elle offre une communication série asynchrone en écriture seule, convertie en USB par un FT232 pour être compatible avec la plupart des ordinateurs récents.
- Elle fournit un emplacement pour une carte mémoire de type *secure digital* (SD) pour une communication par protocole série synchrone (SPI).
- Elle est alimentée sur 4 accumulateurs NiMH pour une tension d'alimentation de 4,8 V et un régulateur linéaire LE33CZ vers 3,3 V, ou une paire d'accumulateurs NiMH et une pompe de charge MAX1674 élevant la tension à 3,3 V.

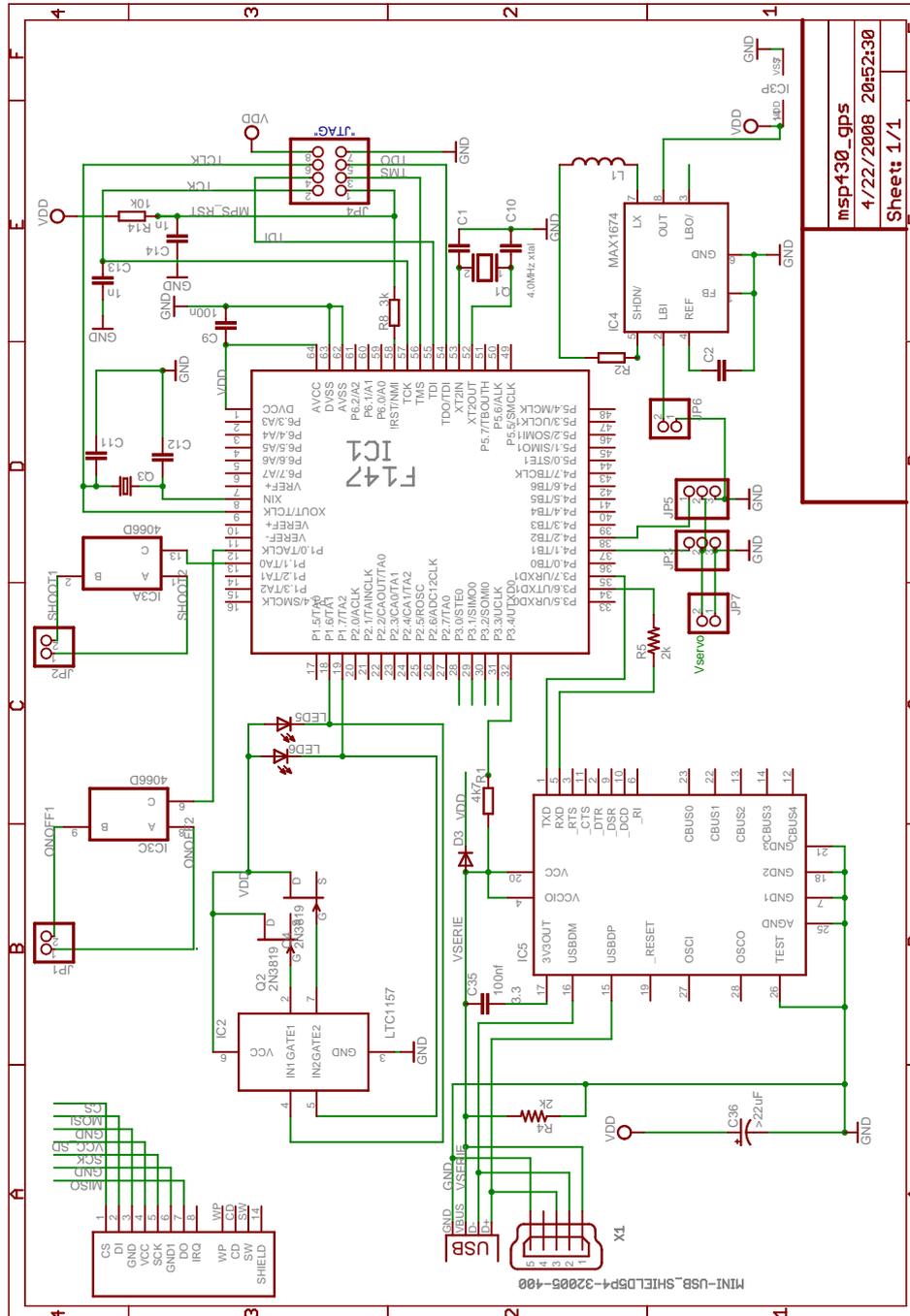


FIG. 5.7 – Schéma électronique de notre plateforme équipée d'un emplacement pour une carte mémoire secure digital (SD) et d'un récepteur GPS

5.4 Description du module GPS

5.4.1 Structure des trames NMEA

Les trames GPS sont des chaînes de caractères en ASCII directement interprétables par un utilisateur et commencent par un \$.

Le protocole de communication du GPS définit 6 types de trames distinctes (Fig. 5.8), chacune ayant une spécificité telle que : l'heure en plus, le nombre de satellites, etc...

```
$GPRMC,101236.000,A,4821.6999,N,00446.5093,W,0.94,167.02,281208,,*11
2 $GPGGA,101237.000,4821.6991,N,00446.5093,W,1,05,8.2,5.1,M,52.5,M,0000*40
$GPRMC,101237.000,A,4821.6991,N,00446.5093,W,0.81,167.67,281208,,*1F
4 $GPGGA,101238.000,4821.6987,N,00446.5092,W,1,05,8.2,5.4,M,52.5,M,0000*4C
```

FIG. 5.8 – Exemple de relevé de trames GPS

5.4.2 Fonctionnement

Le GPS équipant la carte utilise le protocole RS232. Celui-ci utilise le même port que la communication série (Tx pour la communication avec un ordinateur, Rx pour la réception des données GPS). Le second port disponible sur le MSP430 est réservé à l'utilisation du SPI. Le driver exploite directement la couche UART du microcontrôleur, car il y a une incompatibilité avec le système de communication de TinyOS.

En effet, TinyOS offre un mécanisme de communication intégré s'apparentant aux messages TCP/IP classiques : les paquets sont encapsulés dans une trame contenant un certain nombre d'informations pour leur routage et la vérification du contenu. L'intérêt est d'offrir, d'une manière transparente, une interface simple cachant le mode de communication exploité, compatible quelque soit l'endianess³⁷ du microcontrôleur.

Ce modèle possède clairement des avantages, car il permet d'homogénéiser la communication et d'en assurer la fiabilité. Mais cette encapsulation de la communication constitue un handicap dans le cas du GPS qui fournit ses informations au moyen d'une chaîne ASCII d'où l'utilisation de la couche UART.

5.4.3 Module GPS

Comme TinyOS gère les interruptions, il est donc possible d'être averti en permanence de la réception d'un octet sur l'UART. Cela permet de réceptionner des trames et de les stocker en même temps.

Le module fonctionne de la manière suivante :

A la réception du caractère \$, le système stocke les caractères arrivant et quand il détecte le caractère de fin, il envoie à l'application cette chaîne.

Le démarrage et l'arrêt de l'acquisition se fait grâce à l'interface *SplitControl*. Le fonctionnement du module se fait de la manière suivante :

- Lorsque *start()* n'a pas été appelé ou bien que *stop()* a été appelé, l'état est GPS_IDLE. Dans cet état, le module ne s'occupe pas des octets qu'il reçoit.

³⁷L'endianess correspond à l'ordre dans lequel les octets sont organisés en mémoire ou dans une communication

- Après un appel à `start()` ou quand une trame a été envoyée à l'application, le module se met dans l'état `GPS_WAIT`. Dans cet état, il ne stocke rien tant qu'il n'a pas reçu le caractère `$`.
- Lorsqu'il reçoit le caractère de début de trame, il se met dans l'état `GPS_READ` et commence à stocker les caractères qu'il reçoit de l'UART.
- A l'arrivée d'une fin de trame, il passe en `GPS_FINISH` et envoie la chaîne.
- Lorsqu'il a envoyé la chaîne, il se remet en `GPS_WAIT` et attend la réception d'un nouveau caractère `$`.

5.5 Evaluation du module de stockage

Tandis que l'acquisition des données par le GPS se fait à une vitesse de 4800 bauds, nous avons tout d'abord vérifié l'intégrité des données stockées sur la carte SD à l'aide de notre système. Nous avons donc envoyé via le port série RS232 le contenu d'un fichier à une vitesse de 9600 bauds. La comparaison des deux fichiers montre que l'intégrité des données allant jusqu'à 11 MB a été conservée.

L'évaluation de la consommation énergétique est la suivante :

- quand il n'y a aucun accès à la SD, la consommation est d'environ 8 mA (26,4 mW),
- pendant l'activation de la SD et l'écriture en mémoire RAM, la consommation atteint jusqu'à 38 mA (125 mW),
- sur une période à 2 s, nous observons des sauts de consommation allant jusqu'à 60 mA (200 mW), ce qui correspond probablement au transfert des données mises en cache dans la mémoire RAM de la SD sur sa FLASH [138].

Conclusion

Les résultats précédents indiquent que le stockage de données sur mémoire non-volatile est, en terme d'énergie, du même ordre de grandeur que les technologies sans fil à faible consommation et à bas débit telles que Bluetooth et ZigBee avec des temps de réveil plus proches de ceux trouvés pour ZigBee (quelques dizaines de millisecondes) que pour Bluetooth (quelques secondes). Cependant, la bande passante disponible pour le stockage de données est beaucoup plus importante (quelques dizaines de kilo-octets par seconde) que pour la transmission des données par radiofréquence, ce qui donne de plus courtes périodes de réveil et donc diminue la consommation.

Ces premiers résultats montrent l'efficacité de notre système car il conserve l'intégrité des données acquises sur le support de stockage et est compatible avec tous les autres systèmes d'exploitation grâce au système de fichier FAT16 qui y est intégré.

Néanmoins, malgré les avantages précédents, le stockage sur un système de fichiers consomme autant que les technologies sans fil à bas débit qui existent mais pour une bande passante disponible différente. Dans le prochain paragraphe, nous allons étudier et comparer le comportement du stockage d'information par rapport à la transmission via la technologie sans fil : ZigBee.

5.6 Test et performances

5.6.1 Objectif

L'objectif de ce travail est d'étudier le comportement d'une nouvelle architecture de réseau de capteurs au cours d'un déploiement réel. Différentes méthodes telles que le stockage et l'envoi systématique vont être comparées au cours de cette étude. C'est pour cette raison que nous avons dans un premier temps validé la portabilité de notre système de fichier grâce à l'environnement de TinyOS sur plusieurs architectures dotées de différents microcontrôleurs telles que : la plateforme TelosB basée sur un microcontrôleur TI MSP430F611 (Fig. 5.9 (a)) et la plateforme MicaZ s'articulant autour d'un ATmega 128 (Fig. 5.9 (b)), ce qui illustre un des principaux avantages de l'utilisation de TinyOS : la portabilité des environnements logiciels sans aucune dépendance matériels.

Sur la plateforme MicaZ (Fig. 5.9 (b)), la carte SD est reliée aux broches suivantes du connecteur 51 broches de la carte :

- Chip Select est connecté à la broche LED1 (broche 10, PA2),
- MISO est connecté à la broche USART1 RXD (broche 19, PD2),
- MOSI est connecté à la broche USART1 TXD (broche 20, PD3)
- finalement, l'horloge (Clock) est fournie par USART CLK (broche 15, PD5).

Comme l'architecture du microcontrôleur MSP430F1611 de la plateforme TelosB (Fig. 5.9 (a)) est la même que celle utilisée sur notre carte de prototype, la liaison de la carte SD s'effectue en connectant les mêmes broches que dans notre exemple : Chip Select sur P3.0 (broche 28), MOSI sur P3.1 (broche 30), MISO sur P3.2 (broche 30) et l'horloge sur P3.3 (broche 31).

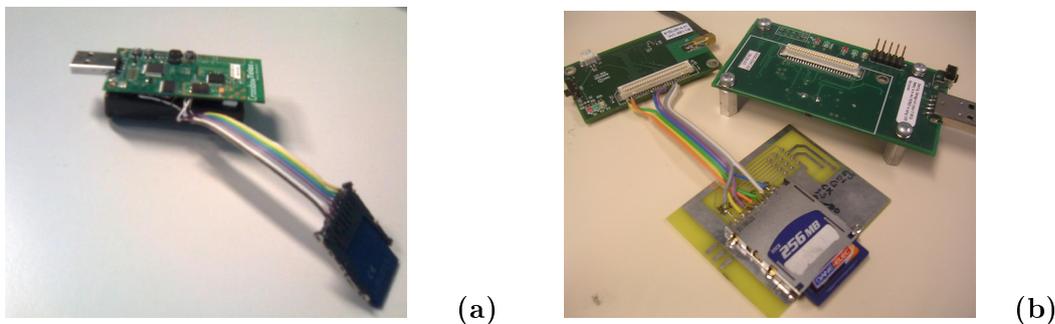


FIG. 5.9 – Validation de la portabilité de notre environnement de stockage sur TelosB (a) et MicaZ (b)

Dans l'expérience qui suit, nous utilisons la TelosB dotée d'une carte SD (Fig. 5.9 (a)) pour le stockage non volatile d'information.

5.6.2 Conditions de l'expérience

Cette expérience s'articule autour de deux points principaux :

1. Estimer la durée de vie des capteurs dans des conditions de vieillissement accéléré,
2. Comparer la transmission d'information et le stockage.

Les conditions expérimentales sont les suivantes :

- 1 carte Fox qui reçoit et traite les données envoyées par différents capteurs,
- 4 capteurs TelosB qui envoient chaque minute la température, la luminosité, la tension des piles,
- 1 capteur TelosB qui stocke toutes les 3 secondes sur une carte SD : un marqueur de temps (Heures : Minutes : Secondes), la température, la luminosité et la tension des piles, et qui envoie également sa tension toutes les 3 heures.

Dans cette expérience où les dernières données ont été reçues après 4 jours et 16 heures, les capteurs ont été dispersés dans une salle de l'UFR de Sciences et Techniques de Besançon comme le montrent les figures 5.10 et 5.11. Les capteurs 1 et 2 sont sur les rebords des fenêtres à chaque bout de la salle, les capteurs 3 et 4 au dessus d'étagères du côté du capteur 2 et le capteur 5, celui équipé de la carte SD pour évaluer le stockage, sur un bureau vers le centre de la salle et près de la carte Fox.

La Fig. 5.12 qui est un plan vu du haut montre que le capteur 1 est le plus éloigné de la carte Fox (12 m) et proche de la fenêtre donc des variations lumineuses ainsi que le capteur 2 (5.5 m). Les capteurs 3 et 4 sont respectivement à 3 et 4 mètres. Puis, le dernier capteur 5 équipé de la carte SD, dont le but n'est pas d'étudier la transmission mais plutôt le stockage, est à 90 cm.



FIG. 5.10 – Photos de la salle (vue de l'entrée)

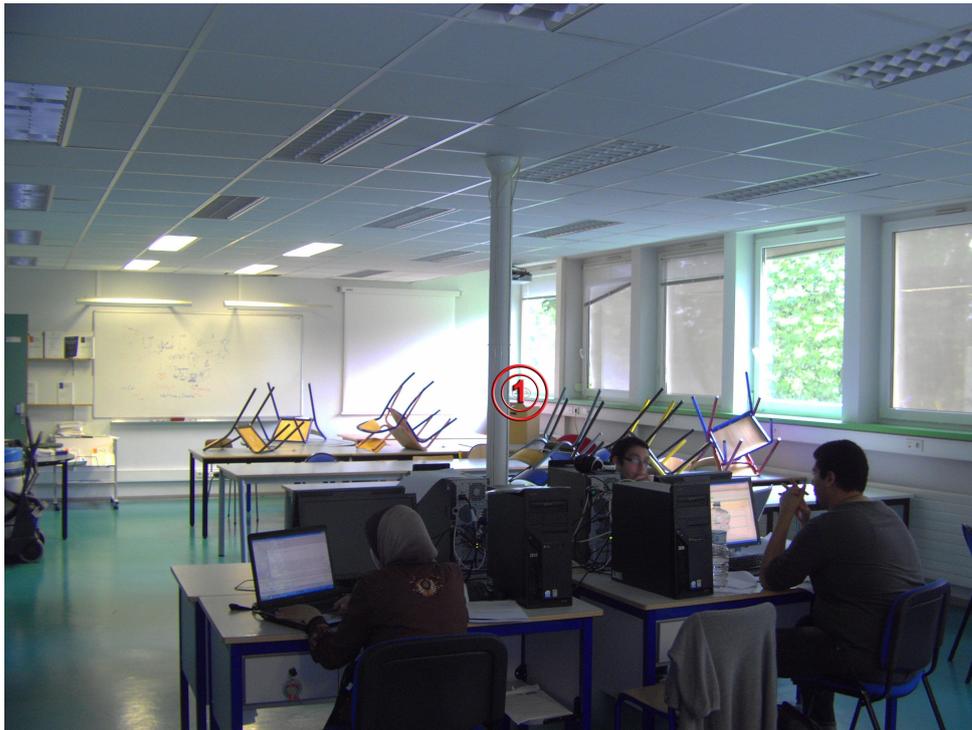


FIG. 5.11 – Photos de la salle (vue du fond)

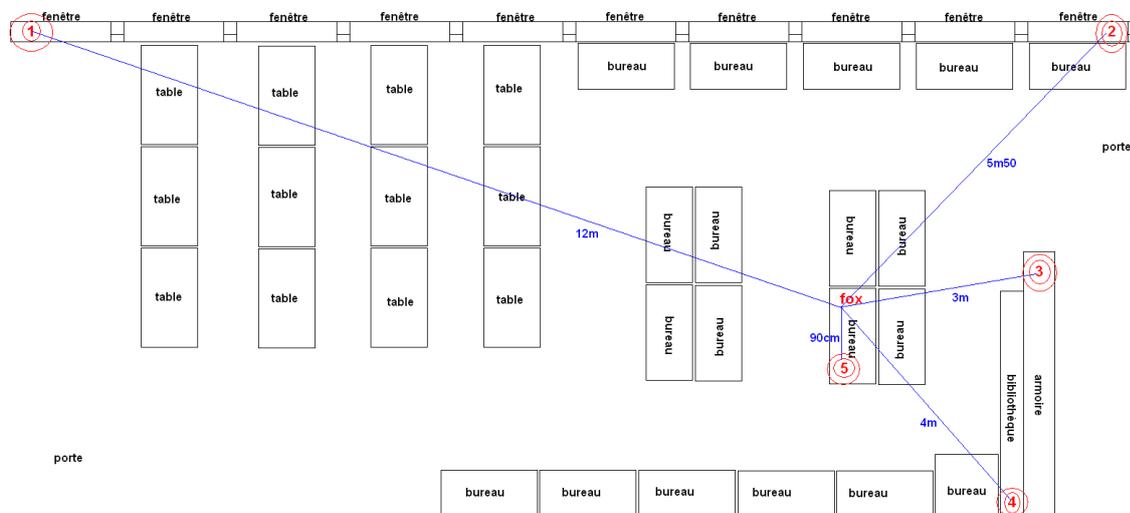


FIG. 5.12 – Plan de la salle vu du haut

5.6.3 Résultats expérimentaux

Comparaison de la consommation énergétique du stockage par rapport à la transmission

Dans un premier temps, nous allons évaluer la consommation de chacun des capteurs (Fig. 5.13). Le capteur 5 équipé de la carte SD a fonctionné correctement pendant 40h. Nous constatons que sur cette période de fonctionnement sans interruption, la tension du capteur 5 décroît

beaucoup moins rapidement que toutes les autres malgré le fait qu'il écrive toutes les 3 secondes sur la carte SD, ce qui montre encore une fois le poids de la transmission.

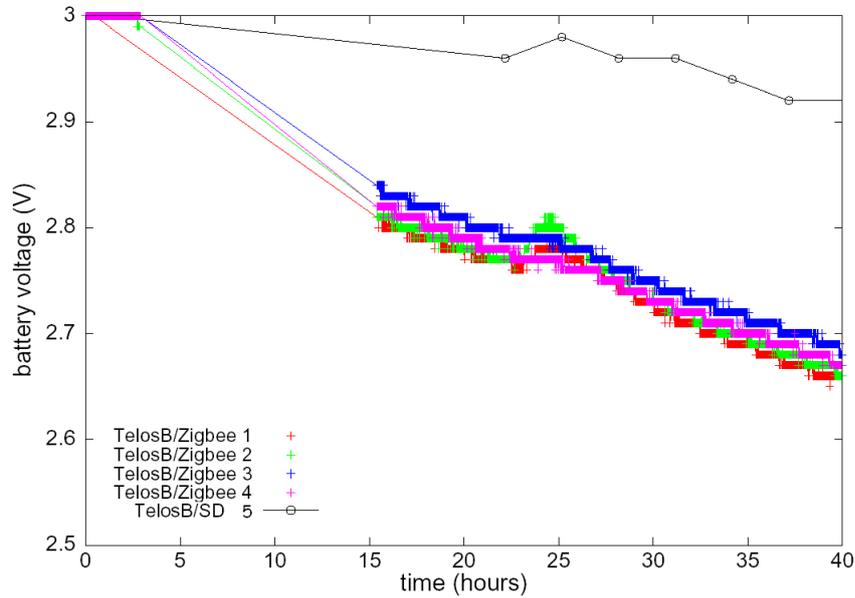


FIG. 5.13 – Variation de la tension des capteurs sur les 40 premières heures

Nous pouvons conclure que pour la même quantité de données, le stockage va consommer moins que l'envoi systématique et va donc permettre de prolonger la vie du réseau.

Étude du comportement des capteurs pendant l'expérience

La Fig. 5.14 illustre les variations des 4 autres capteurs sur toute la durée de l'expérience.

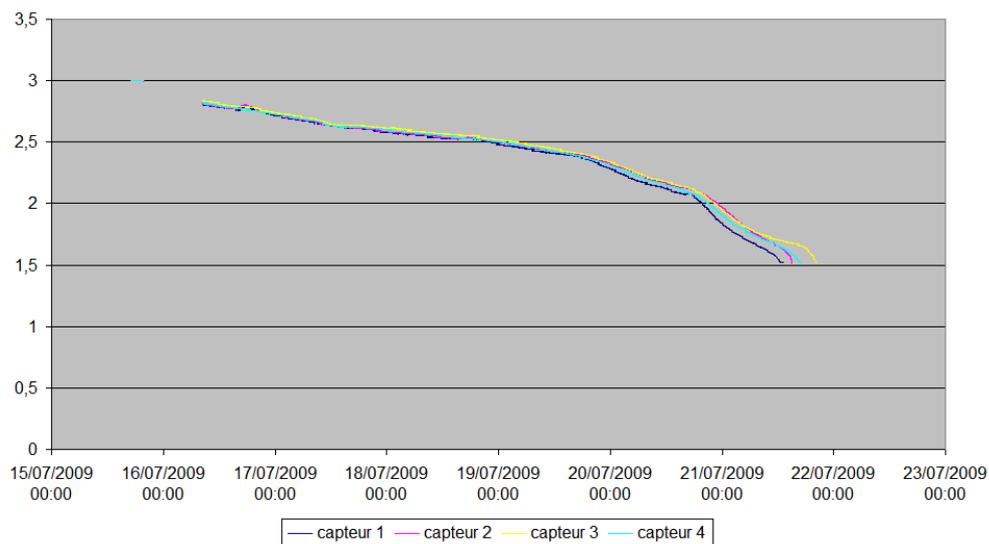


FIG. 5.14 – Variation de la tension des 4 premiers capteurs sur toute la durée de l'expérience

L'étude de ce graphique montre que les courbes des quatre autres capteurs sont surperposables, par contre, la tension des capteurs 1 et 2, qui sont les plus éloignés, décroît plus rapidement. Plus la tension est faible, plus elle décroît vite et aucune donnée n'est transmise quand la tension passe sous la barre de 1,5 V. Nous distinguons mieux l'écart de consommation sous la barre de 2 V.

La Fig. 5.15 représente l'évolution de la température en degrés Celsius. Ce graphique est aussi intéressant du point de vue impact de la variation de tension sur les résultats. En effet, en comparant ce graphique à celui de la tension, nous remarquons qu'en dessous d'un certain niveau d'énergie (2V), la majorité des données sont erronées. En effet, entre le 20 et le 22 juillet, nous observons des températures qui atteignent jusqu'à 250 degrés Celsius dans une salle où la température ambiante ne dépasse pas les 30 degrés Celsius.

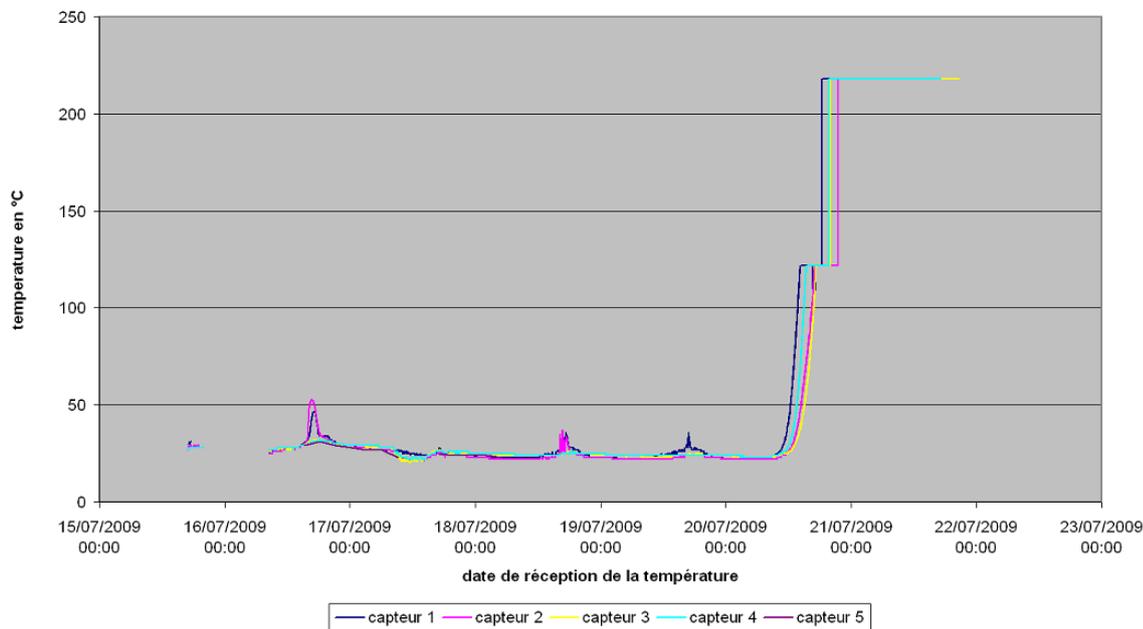


FIG. 5.15 – Variation de la température

Nous pouvons donc en conclure que les capteurs ont besoin d'un niveau minimum d'énergie pour donner des résultats cohérents. En effet, une tension d'alimentation trop faible ne permet pas de définir une tension de référence stable d'où ces résultats.

Un zoom sur les variations précédentes montre que la température des capteurs situés près des fenêtres où il y a plusieurs variations lumineuses atteint jusqu'à 50 degrés Celsius alors que la température de la salle ne dépasse pas les 30 degrés Celsius. Cela concerne les capteurs 1 et 2. Ces variations extrêmes sont dues à l'échauffement par rayonnement direct.

Nous avons donc étudié sur un même graphe les variations de la température et de la luminosité (Fig.5.16) mesurée en lux pour confirmer l'impact du rayonnement direct sur nos résultats. Pour mieux distinguer les variations et ne pas encombrer notre figure, nous n'avons pris en compte que les capteurs 1 et 3 respectivement exposé à un échauffement direct et à l'intérieur

de la salle. La Fig.5.16 montre bien l'influence du rayonnement sur les variations de température.

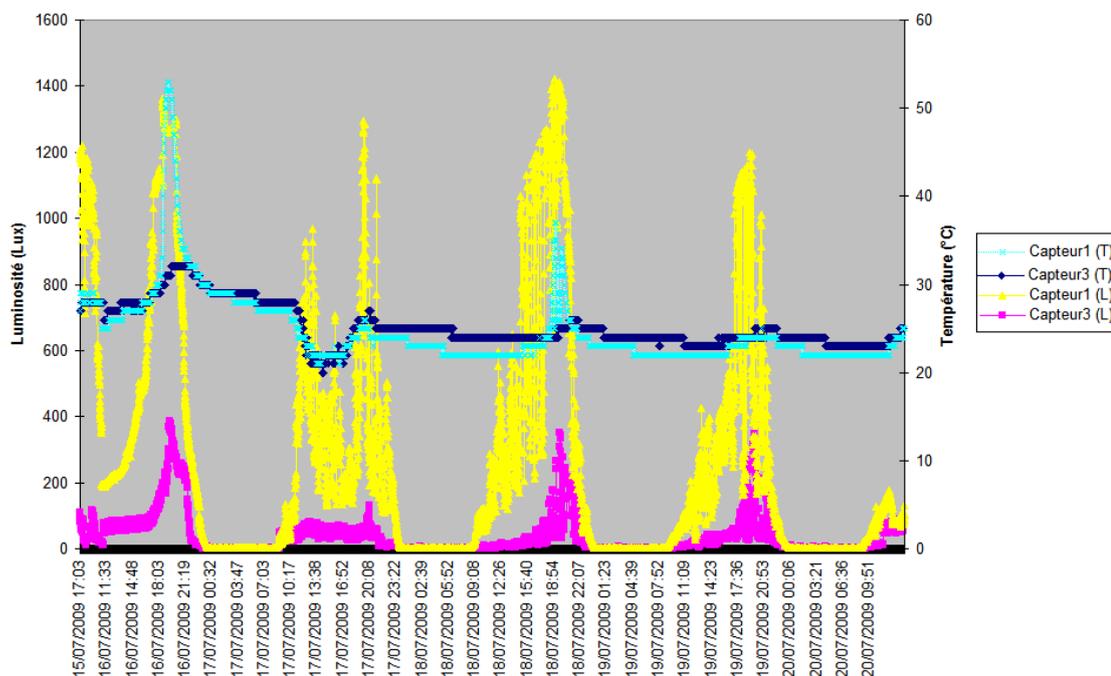


FIG. 5.16 – Variations de la température et de la luminosité

La Fig. 5.17 représente les variations de la luminosité de chaque capteur.

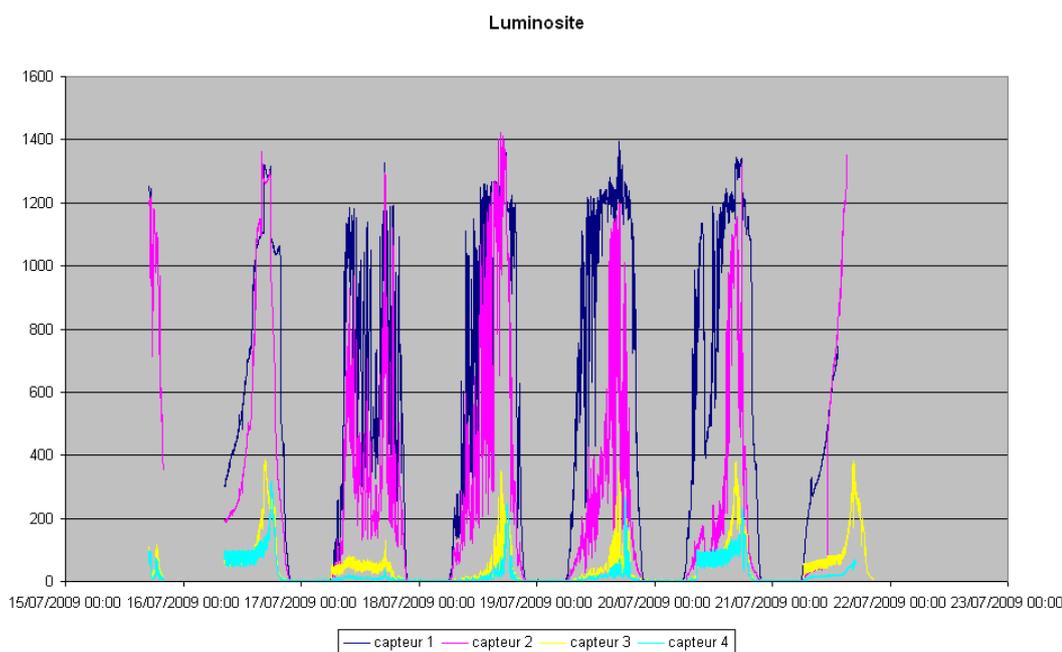


FIG. 5.17 – Variation de la luminosité

La luminosité est mesurée en Lux, (quelques échelles de valeurs : rue éclairée la nuit 20 à 40 lux, appartement bien éclairé 200 à 400 lux, extérieur par temps couvert 25 000 lux, extérieur par temps ensoleillé 50 000 à 100 000 lux).

Nous constatons bien sur ce graphique les différences jour/nuit et nous distinguons bien les capteurs aux bords des fenêtres des capteurs situés dans la salle (avoisinant 1400 lux dans la journée pour les premiers et ne dépassant pas 400 lux durant cette même période pour les seconds).

En comparant ce graphique à celui de la tension, nous constatons que les capteurs vers les fenêtres consomment légèrement plus que ceux à l'ombre mais ils sont également placés à une plus longue distance.

5.7 Conclusion

Les résultats qui précèdent ont mis en valeur l'intérêt du stockage par rapport à la transmission. Nous constatons qu'après 40h de fonctionnement, le capteur qui stocke les informations a une tension autour de 2,9 V tandis que la tension des quatre autres capteurs qui transmettent systématiquement est inférieure à 2,65 V. Cette différence de 0,25 V montre l'avantage du stockage par rapport à la transmission.

Un autre constat important à prendre en compte est l'incohérence des données collectées à partir d'une tension en dessous de 2 V.

L'influence de la distance sur la consommation est également mise en évidence car les capteurs les plus éloignés vident leur batterie plus rapidement.

Conclusion et perspectives

Conclusion

Les réseaux de capteurs, nouvelle thématique de recherche innovante, vient enrichir le domaine des réseaux et télécommunications. La dimension de ce nouveau type de réseau capable de collecter des informations sur un environnement et de les stocker ou les transmettre à une station collectrice, va permettre de surveiller différentes zones qui peuvent être hostiles à l'homme. La pluridisciplinarité de ce domaine pousse à la convergence des compétences de plusieurs laboratoires qui ont parfois des centres d'intérêts différents mais qui se rejoignent sur cette thématique. La particularité de ces nouveaux systèmes distribués émergents est la capacité réduite des nœuds en terme de calcul, de mémoire, ou d'énergie. Cette dernière caractéristique est le point critique dans ces réseaux. Les recherches dans ce domaine se focalisent donc dans la majorité des cas sur la gestion énergétique.

Dans cette thèse, nous avons présenté une architecture permettant d'interconnecter plusieurs capteurs hétérogènes répartis sur deux niveaux hiérarchiques. Cette architecture est capable de fournir des informations riches à l'utilisateur final en prenant en compte les contraintes énergétiques grâce à son niveau le plus haut capable d'effectuer des tâches complexes qui sont difficiles au niveau plus bas. Cette combinaison de plateformes offre donc plus de fonctionnalités à notre architecture et un champ d'application plus vaste. Nous avons étudié les plateformes qui composent cette architecture ainsi que les technologies sans fil d'interconnexion entre les différents capteurs.

Pour optimiser la collecte et le transfert de données dans la zone de déploiement de ce réseau de capteurs, nous avons proposé deux techniques :

- Une compression de données permettant de réduire la taille des informations transmises tout en respectant la cohérence des données et nécessitant une faible consommation énergétique. Les deux algorithmes de compression de données proposés dans le cadre de cette thèse, K-RLE et Aganmà, comparés à l'algorithme de référence dans le domaine, ont montré leur efficacité en terme de taux de compression et de consommation énergétique. Dans un premier temps, nous avons proposé un algorithme de compression de données avec perte, s'inspirant de l'algorithme RLE, en introduisant un nouveau paramètre K. Le choix de ce paramètre a fait l'objet d'une étude à l'aide de deux principaux outils : l'écart-type et la variance d'Allan. Puis, nous avons introduit un second algorithme de compression de données sans perte capable de s'adapter en fonction des variations de températures.
- Un stockage de données sur carte amovible de type SD. Pour ce faire, nous avons introduit sous TinyOS-2.x un système de fichier FAT16. Ce nouvel environnement va non seulement permettre une observation dans des conditions environnementales difficiles à la radiofréquence ou encore le stockage d'une quantité importante d'information mais aussi, il ne nécessite aucun post-traitement pour l'utilisateur final qui pourra directement utiliser son support de stockage sur n'importe quel système d'exploitation.

Ces deux propositions implémentées sur une architecture hétérogène sont de nouveaux outils permettant d'utiliser un réseau de capteurs avec des qualités supérieures en terme de performance et de consommation d'énergie.

Cette thèse a permis de lever certains verrous de recherche dans le domaine des réseaux de capteurs et a montré la richesse de cette thématique. Les différentes problématiques abordées et la multitude d'applications qui peuvent y être rattachées ou encore la pluridisciplinarité de

cette thématique lui procure un avenir prometteur.

Perspectives

Dans cette thèse, nous avons présenté une architecture hiérarchique de réseaux de capteurs utilisable dans différents contextes. L'étude des technologies sans fil dédiées aux réseaux de capteurs nous a poussé à mettre l'accent sur le standard IEEE 802.15.4 et plus précisément sur le compromis gain énergétique/délai qui lui est associé. Le résultat de cette étude met l'accent sur la difficulté de considérer des applications avec des contraintes fortes en terme de temps et nous a poussé à nous focaliser sur des applications d'observation mais il serait intéressant avec l'introduction de l' α -pénalité de déterminer théoriquement une période de veille en fonction d'une application précise et de valider ce choix de manière expérimentale.

Les applications basées sur des observations sans contraintes temporelles permettent l'optimisation du canal de communication en utilisant des techniques de compression de données. Nous avons proposé deux nouvelles techniques de compression de données : K-RLE et Aganmà. La caractérisation du paramètre K de l'algorithme K-RLE reste encore une voie intéressante à explorer car il influe sur la qualité des données décompressées. Dans ce travail, nous avons utilisé l'écart-type et la variance d'Allan comme techniques pour caractériser K en fonction des variations du lieu où les capteurs sont sensés être déployés mais il serait intéressant de caractériser K à l'aide d'autres méthodes efficaces pour éliminer le bruit en minimisant la perte de données. Aganmà qui reste dans cette étude le meilleur compromis entre qualité de données et fort taux de compression nécessite par contre un temps de compression considérable par rapport aux autres algorithmes comparés. Il serait judicieux alors d'optimiser le temps de compression de cet algorithme. Le choix de subdivision de l'espace de compression d'un octet en deux étant arbitraire, on pourrait également mieux le caractériser en fonction des variations de températures en utilisant l'écart-type par exemple.

Le nouvel environnement de stockage d'informations développé sous TinyOS-2.x en utilisant le système de fichiers FAT16 sur support amovible facilite l'analyse à l'utilisateur final. Par contre, le souhait de garantir une cohérence des données dans le temps lui procure une consommation énergétique proche de celle des technologies sans fil à bas débit dédiées aux réseaux de capteurs. Il serait donc intéressant d'optimiser la consommation énergétique de cet environnement tout en conservant la cohérence des données. Une autre voie serait également à explorer. Elle concerne la comparaison en terme d'intégrité des données, de consommation énergétique et de bande passante offerte de notre environnement de stockage d'informations utilisant le système de fichiers FAT16 avec d'autres environnement développés pour les plateformes Shimmer/Shimmer2 et dotées de ce même système de fichiers. Cette comparaison nécessitera tout d'abord une mise à jour de l'environnement développé pour les plateformes Shimmer.

Bibliographie

- [1] Thèse - Yann Busnel. Systèmes d'information collaboratifs et auto-organisés pour réseaux de capteurs large-échelle : " De la théorie à la pratique ". Université de Rennes 1, Rennes, France, Novembre 2008.
- [2] (2003-2008) Napster, LLC. [Online]. Available : <http://free.napster.com/>
- [3] The E-Mule project. [Online]. Available : <http://www.emule-project.net/>
- [4] (2001-2009) The BitTorrent website. [Online]. Available : <http://bittorrent.com/>
- [5] (2009) Skype official website. [Online]. Available : <http://www.skype.com/>
- [6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, Wireless sensor networks : a survey, *Computer Networks*, vol. 38, no. 4, pp. 393-422, 2002.
- [7] P. Levis, and S. Madden, and J. Polastre, and R. Szewczyk, and K. Whitehouse, and A. Woo, and D. Gay, and J. Hill, and M. Welsh, and E. Brewer, and D. Culler, TinyOS : An Operating System for Sensor Networks, *Journal Ambient Intelligence*, pp. 115-148, 2005.
- [8] M. Healy, T. Newe, and E. Lewis. Power management in operating systems for wireless sensor nodes. *IEEE Sensors Applications Symposium, 2007 (SAS '07)*, pp. 1-6, San Diego, California, USA, Feb. 2007.
- [9] A. Dunkels, B. Grönvall, and T. Voigt. Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. In *Proc. of the First IEEE Workshop on Embedded Networked Sensors*, pp. 455-462, Tampa, Florida, USA, November 2004.
- [10] (2008) The Crossbow website. [Online]. Available : <http://www.xbow.com>
- [11] D. Gay, P. Levis, R. Von Behren, M. Welsh, E. Brewer, and D. Culler, The nesC Language : A Holistic Approach to Networked Embedded Systems, In *Proceedings of Programming Language Design and Implementation (PLDI 2003)*, pp. 1-11, San Diego, California, USA, June 2003.
- [12] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han. Mantis os : an embedded multithreaded operating system for wireless micro sensor platforms. *Mob. Netw. Appl.*, Vol. 10, No. 4, pp. 563-579, 2005.
- [13] C. C. Han, R. Kumar, R. Shea, E. Kohler, and M. Srivastava. A dynamic operating system for sensor nodes. In *MobiSys '05 : Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pp. 163-176, New York, USA, 2005.
- [14] <http://www.ethernut.de/>
- [15] (2007) The BTnode Project. [Online]. <http://www.btnode.ethz.ch/>
- [16] The Acme Systems website. Available : <http://www.acmesystems.it>
- [17] <http://nesc.sourceforge.net/papers/nesc-ref.pdf>
- [18] (2004) The TinyOS Community Forum. <http://www.tinyos.net/>

- [19] N. Kimura and S. Latifi. A survey on data compression in wireless sensor networks. In *International Conference on Information Technology : Coding and Computing*, volume 2, pp. 8–13, 2005.
- [20] Thèse - VAN DEN BOSSCHE Adrien - Proposition d'une nouvelle méthode d'accès déterministe pour un réseau personnel sans fil à fortes contraintes temporelles, Spécialité : Génie Informatique, Automatique et Traitement du Signal, le 06 Juillet 2007.
- [21] M. Augel and R. Knorr. Bluetooth scatternet formation - state of the art and a new approach. In *ARCS*, pp. 260–272, Augsburg, Germany, 2004.
- [22] G. V. Zaruba, S. Basagni, and I. Chlamtac. Bluetrees-scatternet formation to enable bluetooth-based ad hoc networks. In *IEEE International Conference on Communications, 2001. ICC 2001.*, volume 1, pp. 273–277, St. Petersburg, Russia, 2001.
- [23] Z. Wang, R. Thomas, and Z. Haas. Bluenet – a new scatternet formation scheme. *Hawaii International Conference on System Sciences*, Vol. 2, pp. 61, Hawaii, 2002.
- [24] (2009) Bluetooth SIG - The Official Wibree website. <http://www.wibree.com/>
- [25] J. Lansford - HomeRF : Bringing Wireless Connectivity Home - Intel HomeRF technology Tutorial, Avril 1999.
- [26] IEEE-TG15.4, *Part 15.4 : Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE standard for Information Technology, 2003.
- [27] (2009) The ZigBee Alliance website. [Online]. Available : <http://www.zigbee.org/>
- [28] M. A. Hanson, H. C. P. Jr., A. T. Barth, K. Ringgenberg, B. H. Calhoun, J. H. Aylor, and J. Lach. Body area sensor networks : Challenges and opportunities. *Computer*, Vol. 42, No. 1, pp. 58–65, 2009.
- [29] Cours de Yacine CHALLAL - Université de Technologie, Compiègne. http://tice.utc.fr/moodle/file.php/498/SupportWeb/co/Module_RCSF_60.html/
- [30] H. Labiod, editor. *Réseaux mobiles ad hoc et réseaux de capteurs sans fil*. Hermes Science Publications - Traité IC2, série Réseaux et télécommunications, ISBN 2-7462-1292-7, Avril 2006.
- [31] John Jubin and Janet D. Tornow. The DARPA Packet Radio Network Protocols. *Proceedings of the IEEE*, 75(1), January 1987.
- [32] Haas, Z.J., Pearlman, M.R. and Samar, P., Zone Routing Protocol (ZRP), IETF Internet Draft, draft-ietf-manet-zrp-04.txt, Juillet 2002.
- [33] C.-C. Yang and L.-P. Tseng. Fisheye zone routing protocol : A multi-level zone routing protocol for mobile ad hoc networks. *Comput. Commun.*, Vol. 30, No. 2, pp. 261–268, 2007.
- [34] A. Iwata, g. Ching-chuan Chian, G. Pei, M. Gerla, and T. wei Chen. Scalable routing strategies for ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, Vol. 17, pp. 1369–1379, 1999.
- [35] B. Karp and H. T. Kung. Gpsr : greedy perimeter stateless routing for wireless networks. In *MobiCom '00 : Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 243–254, New York, USA, 2000.
- [36] Y.-B. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, Vol. 6, Issue 4, pp. 307–321, July 2000.
- [37] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. In *SIGCOMM*, pp. 234–244, 1994.

-
- [38] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr), 2003.
- [39] R. Ogier, F. Templin, M. Lewis, Topology Dissemination Based on Reverse-Path Forwarding (TBRPF), IETF Network Working Group RFC3684, February 2004.
- [40] D. B. Johnson, D. A. Maltz, and J. Broch. Dsr : The dynamic source routing protocol for multi-hop wireless ad hoc networks. *In Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5*, pp. 139–172, 2001.
- [41] C. Perkins, E. Belding-Royer, S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, IETF Network Working Group RFC3561, July 2003.
- [42] Thèse - Mohammed Lehsaini, Diffusion et couverture basées sur le clustering dans les réseaux de capteurs. Thèse en Cotutelle : Université A.B Tlemcen (Algérie) et Université de Franche-Comté (France), juillet 2009.
- [43] J. A. Paradiso and T. Starner. Energy scavenging for mobile and wireless electronics. *Pervasive Computing, IEEE*, Vol. 4, No. 1, pp. 18–27, 2005.
- [44] W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. *In MobiCom '99 : Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 174–185, New York, USA, 1999.
- [45] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion : a scalable and robust communication paradigm for sensor networks. *In MobiCom '00 : Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 56–67, New York, USA, 2000.
- [46] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. *In Proceedings of Hawaii International Conference on System Sciences*, pp. 3005-3014, Hawaii, 2000.
- [47] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, pp. 660–670, 2002.
- [48] D. Braginsky and D. Estrin, Rumor Routing Algorithm for Sensor Networks, in the Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA, October 2002.
- [49] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks : a survey. *IEEE Wireless Communications*, Vol. 11, No. 6, pp. 6–28, 2004.
- [50] K. Akkaya and M. Younis, A Survey of Routing Protocols in Wireless Sensor Networks. *In the Elsevier Ad Hoc Network Journal*, Vol. 3/3, pp. 325-349, 2005.
- [51] J. Carle D. Simplot. Energy efficient area monitoring by sensor networks. IEEE computer, pp. 40-46, New York, USA, February 2004.
- [52] J. Pottie, and W. J. Kaiser, Wireless Integrated Network Sensors, *Communications of the ACM*, Vol. 43, No. 5, pp. 51–58, May 2000.
- [53] I. Bekmezci and F. Alagoz. New tdma based sensor network for military monitoring (mil-mon). *In Military Communications Conference, 2005. MILCOM 2005. IEEE*, Vol. 4, pp. 2238–2243, New Jersey, USA, Oct. 2005.
- [54] A. Milenković, C. Otto, and E. Jovanov. Wireless sensor networks for personal health monitoring : Issues and an implementation. *Computer Communications (Special issue : Wireless Sensor Networks : Performance, Reliability, Security, and Beyond)*, Vol. 29, pp. 2521–2533, 2006.

- [55] H.Y. Zhou, K.M. Hou, J. Ponsonnaille, L. Gineste, C. de Vault, A New System Dedicated to Real-time Cardiac Arrhythmias Tele-assistance and Monitoring, *Journal of Universal Computer Science*, Vol. 12, No. 1, pp. 30-44, 2006.
- [56] N. Kurata, Jr B. F. Spencer, Ruiz-Sandoval M. Risk monitoring of buildings with wireless sensors network. *Structural Control and Health Monitoring*, Vol. 12, pp. 315-327, 2005.
- [57] E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, and V.Z. Groza. Sensor-based information appliances. *IEEE Instrumentation Measurement Magazine*, Vol.3, No.4, pp. 31-35, December 2000.
- [58] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, M. Welsh. Deploying a Wireless Sensor Network on an Active Volcano, *IEEE Internet Computing*, Vol. 10, No. 2, pp. 18–25, March 2006.
- [59] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong. A macroscope in the redwoods. *In SenSys '05 : Proceedings of the 3rd international conference on Embedded networked sensor systems*, pp. 51–63, New York, USA, 2005.
- [60] Martinez, K., Ong, R. and Hart, J. (2004) Glacsweb : a sensor network for hostile environments. *In Proceedings of The First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, Santa Clara, USA, 2004.
- [61] F. Ingelrest, G. Barrenetxea and M. Vetterli. SensorScope, un système clef en main de surveillance de l'environnement. 13ème Colloque Francophone sur l'Ingénierie des Protocoles (CFIP 2008). Les Arcs, France, 25 - 28 March 2008.
- [62] (2006-2010) le programme Hydor-sensor-flow. <http://www.hydro-sensor-flow.com/>
- [63] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges : Scalable coordination in sensor networks. *In Proc. ACM/IEEE MobiCom*, pp. 263-270, Seattle, Washington, United States, 1999.
- [64] D. Estrin. Embedded networked sensing research : Emerging systems challenges. *In NSF Workshop on Distributed Communications and Signal Processing*. Northwestern University, Evanston, Illinois, USA, December 2002.
- [65] Vincent Lecuire - Présentation GDR MACS - Paris, Mars 2009.
- [66] Jason Hill's Home page. http://www.jhlhlab.com/jhill_cs/
- [67] J. Polastre, R. Szewczyk, D. E. Culler : Telos : enabling ultra-low power wireless research. *In Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pp. 364-369, Los Angeles, California, USA, 2005.
- [68] The Sensor Museum. <http://www.btnode.ethz.ch/Projects/SensorNetworkMuseum/>
- [69] <http://www.snm.ethz.ch/Projects/AtmelAVR/>
- [70] http://www.btnode.ethz.ch/pub/uploads/Projects/CC2420_datasheet.pdf
- [71] Jason Hill, Mike Horton, Ralph Kling, Lakshman Krishnamurthy. The platforms enabling wireless sensor networks, *Communications of the ACM*, Volume 47, pp. 41-46 , June 2004.
- [72] 1995-2009 The Texas Instruments website. <http://focus.ti.com/>
- [73] J. Beutel, O. Kasten, F. Mattern, K. Römer, F. Siegemund and L. Thiele : Prototyping Wireless Sensor Networks with BTnodes. *Proc. 1st European Workshop on Wireless Sensor Networks (EWSN 2004)*, Springer LNCS, vol. 2920, pp. 323-338, Berlin, Germany, January 2004.

-
- [74] <http://www.mics.ch/>
- [75] <http://www.smart-its.org/>
- [76] (2009) <http://sunspotworld.com/products/>
- [77] B. Warneke, M. Last, B. Liebowitz, and K. Pister. Smart Dust : Communicating with a Cubic-Millimeter Computer. *IEEE Computer*, Vol. 34, No. 1, pp. 44-51, 2001.
- [78] O. Kasten, M. Langheinrich. First Experiences with Bluetooth in the Smart-Its Distributed Sensor Network. *In Workshop on Ubiquitous Computing and Communications*, Barcelona, Spain, Sept. 2001.
- [79] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking : design tradeoffs and early experiences with zebanet. *In ASPLOS-X : Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, volume 37, pp. 96–107, New York, USA, October 2002.
- [80] (2004 - 2009) <http://prisms.cs.umass.edu/dome/turtlenet>
- [81] P. Zhang, C. Sadler, S. Lyon, and M. Martonosi, Hardware Design Experiences in ZebraNet, *In Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, Baltimore, MD, USA, November 2004.
- [82] C. M. Sadler and M. Martonosi. Data compression algorithms for energy-constrained devices in delay tolerant networks. *In Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys 2006)*, pp. 265–278, Boulder, Colorado, USA, 2006.
- [83] T. A. Welch. A technique for high-performance data compression. *Computer*, 17(6) :8–19, 1984.
- [84] G. Mathur, P. Desnoyers, D. Ganesan, and P. J. Shenoy. Capsule : an energy-optimized object storage system for memory-constrained sensor devices. *In Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys 2006)*, pp. 195–208, Boulder, Colorado, USA, 2006.
- [85] The Tinynode website. <http://www.tinynode.com/>
- [86] P. Buonadonna, D. Gay, J. M. Hellerstein, W. Hong, and S. Madden. Task : Sensor network in a box. *In Proceedings of European Workshop on Sensor Networks*, pp. 133–144, Istanbul, Turkey, 2005.
- [87] The Sensorscope website. http://sensorscope.epfl.ch/index.php/Network_Code/
- [88] J. Rabaey, J. Ammer, J. Da Silva Jr., D. Patel, and S. Roundy. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *Computer*, Vol. 33, No. 7, pp. 42-48, July 2000.
- [89] R. Min, M. Bhardwaj, S.-H. Cho, N. Ickes, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan. Energy-Centric Enabling Technologies for Wireless Sensor Networks. *IEEE Wireless Communications*, 9(4) :pp. 28-39, August 2002.
- [90] C. C. Enz, A. E. Hojydi, J. D. Decotignie, and V. Peiris. Wisenet : An ultralow-power wireless sensor network solution. *Computer*, Vol. 37, No. 8, pp. 62–70, 2004.
- [91] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless Sensor Networks for Habitat Monitoring. *In Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, Georgia, USA, 2002.
- [92] The COMMON-Sense Net website. <http://commonsense.epfl.ch/>

- [93] I. Talzi, A. Hasler, S. Gruber, and C. F. Tschudin. Permasense : investigating permafrost with a wsn in the swiss alps. *In the Fourth Workshop on Embedded Networked Sensors*, pp. 8–12, Cork, Ireland, 2007.
- [94] Prof. Matt Welsh, Course Introduction to WSN. <http://www.eecs.harvard.edu/~mdw/course/cs263/intro.pdf>, January 29, 2009.
- [95] K. Martinez, P. Basford, J. Ellul and R. Spanton. Gumsense - a high power low power sensor node. Accepted Poster. *In the 6th European Conference on Wireless Sensor Networks (EWSN 2009)*, Cork, Ireland, February 2009.
- [96] Laboratoire d’Informatique, de Modélisation et d’Optimisation des Systèmes de Clermont-Ferrand. <http://www.isima.fr/smir/pages/themes/themes.php?id=1>
- [97] M. H. Rahimi, R. Baer, O. I. Iroezi, J. C. García, J. Warrior, D. Estrin, and M. B. Srivastava, Cyclops : in situ image sensing and interpretation in wireless sensor networks. *In Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys 2005)*, pp. 192-204, San Diego, California, USA, 2005.
- [98] J. Hurwitz, S. Smith, A. Murray, P. Denyer, J. Thomson, S. Anderson, E. Duncan, A. Kinsey, B. Paisley, P.-F. Pugibet, E. Christison, B. Laffoley, M. Panaghiston, S. Bradshaw, J. Vittu, R. Brechignac, and K. Findlater, A miniature imaging module for mobile applications, *IEEE International Solid-State Circuits Conference (ISSCC 2001)*. Digest of Technical Papers. pp. 90-91, 436, San Francisco, California, USA, 2001.
- [99] A. El Gamal, Trends in CMOS Image Sensor Technology and Design, *International Electron Devices Meeting Digest of Technical Papers*, pp. 809-812, 2002.
- [100] W. chi Feng, B. Code, E. C. Kaiser, M. Shea, W. chang Feng, and L. Bavoil, Panoptes : scalable low-power video sensor networking technologies. *In ACM Transactions on Multimedia Computing, Communications, and Applications*, pp. 151-167, 2005.
- [101] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, A survey on wireless multimedia sensor networks, *Computer Networks*, vol. 51, no. 4, pp.921-960, 2007.
- [102] P. Kulkarni, D. Ganesan, P. J. Shenoy, and Q. Lu, SensEye : a multi-tier camera sensor network. *In Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 229-238, 2005.
- [103] P. Kulkarni, D. Ganesan, and P. J. Shenoy. The case for multi-tier camera sensor networks. *In International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV 2005)*, pp. 141–146, Skamania, Washington, USA, 2005.
- [104] E. Ardizzone, M. L. Cascia, G. L. Re, and M. Ortolani. An integrated architecture for surveillance and monitoring in an archaeological site. *In VSSN '05 : Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, pp. 79–86, New York, USA, 2005.
- [105] M. Ortolani, L. Gatani, G. Lo Re, A. Urso, and S. Gaglio. An efficient retransmission strategy for data gathering in wireless sensor networks. *In Proc. of IEEE ETFA05*, Catania, Italy, Sep. 2005.
- [106] <http://developer.axis.com/>
- [107] R. Kapoor, M. Y. M. Sanadidi, and M. Gerla. An analysis of bluetooth scatternet topologies. *In IEEE International Conference on Communications (ICC 2003)*, pp. 266–270, Anchorage, Alaska, USA, 2003.

-
- [108] J. Beutel, M. Dyer, L. Meier, and L. Thiele, Scalable topology control for deployment-sensor networks. In *Proc. 4th Intl Conf. Information Processing in Sensor Networks (IPSN 05)*, pp. 359-363, UCLA, Los Angeles, California, USA, Apr. 2005.
- [109] M. Leopold, M. B. Dydensborg, and P. Bonnet. Bluetooth and sensor networks : a reality check. In *SenSys '03 : Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 103-113, New York, USA, 2003.
- [110] (2000-2008) The Official Linux Bluetooth protocol stack website. [Online]. Available : <http://www.bluez.org>
- [111] (2007) the official site of the Joint Photographic Experts Group. Available : <http://www.jpeg.org/>
- [112] D. Taubman. High performance scalable image compression with ebcot. *IEEE Transactions on Image Processing*, Vol. 9, No. 7, pp. 1158-1170, 2000.
- [113] (2007-2009) The sentilla website. <http://www.sentilla.com/>
- [114] Jérôme Vernez - Projet de diplôme - Adaptabilité de la couche MAC du standard IEEE 802.15.4 à une couche physique Ultra Wide Band, CSEM, 2003.
- [115] J. Yoon, H. Kim, and J.-G. Ko. Data fragmentation scheme in iee 802.15.4 wireless sensor networks. In *IEEE 65th Vehicular Technology Conference (VTC 2007 Spring)*, pp. 26-30, Dublin, Ireland, 2007.
- [116] E. Pamba Capo-Chichi, H. Guyennet, J.-M. Friedt, I. Johnson, and C. Duffy. Design and implementation of a generic hybrid wireless sensor network platform. In *Proceedings of the 33rd IEEE Conference on Local Computer Networks (LCN), WLN*, pp. 836-840, Montreal, Canada, 2008.
- [117] E. Pamba Capo-Chichi, H. Guyennet, and J.-M. Friedt. Ieee 802.15.4 performance on a hierarchical hybrid sensor network platform. In *Proceedings of the Fifth International conference on Networking and Services, ICNS 2009*, 303-308, Valencia, Spain, 2009.
- [118] A. Cunha, A. Koubaa, R. Severino, and M. Alves. Open-zb : an open-source implementation of the iee 802.15.4/zigbee protocol stack on tinyos. In *4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'07)*, pp. 1-12, Pisa, Italy, 2007.
- [119] (2009) The Silicon Laboratories. <https://www.silabs.com/>
- [120] M. Neugebauer, J. Plonnigs, and K. Kabitzsch. A new beacon order adaptation algorithm for iee 802.15.4 networks. In *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN 2005)*, pp. 302-311, Istanbul, Turkey, 2005.
- [121] Nicolas Salles et Nicolas KROMMENACKER. Analyse de performances de la période sans contention de IEEE 802.15.4 pour des applications industrielles temps réel. 5ème Ecole d'été Temps Réel, Nantes, France, 2007.
- [122] F. Marcelloni and M. Vecchio. A simple algorithm for data compression in wireless sensor networks. *IEEE Communications Letters*, Vol. 12, No. 6, pp. 411-413, June 2008.
- [123] D. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, Vol. 40, No. 9, pp. 1098-1101, Sept. 1952.
- [124] D. Salomon. *Data Compression : The Complete Reference*. Second edition, 2004.
- [125] http://www.fileformat.info/mirror/egff/ch09_03.htm
- [126] E. Pamba Capo-Chichi, H. Guyennet, and J.-M. Friedt. K-rlc : A new data compression algorithm for wireless sensor network. In *Proceedings of the Third International Conference on Sensor Technologies and Applications, SENSORCOMM 2009*, pp. 502-507, Athens/Glyfada, Greece, 2009.

- [127] The Weather Underground website. [Online]. Available : <https://english.underground.com>.
- [128] Texas Instruments MSP430x1xx Family User's Guide, 2006. [Online]. Available : <http://focus.ti.com/>
- [129] A. Fraboulet, G. Chelius, and E. Fleury. Worldsens : development and prototyping tools for application specific wireless sensors networks. In *Proceedings of the 6th international conference on Information processing in sensor networks (IPSN 2007)*, pp. 176–185, Cambridge, Massachusetts, USA, 2007.
- [130] D. W. Allan. Time and frequency (time-domain) characterization, estimation, and prediction of precision clocks and oscillators. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 34(6) :647–654, 1987.
- [131] F. Busque, S. Virally, J.S. Corbeil, D.A. Guzman, F. Beaudet, M. LEDUC, V. Couderc Vincent, P. Leproux, G. Huss, N. Gobout, S. Lacroix - Utilisation de la variance d'Allan pour caractériser la stabilité en puissance de sources pulsées, Journées Nationales d'Optique Guidée, Poster, Lannion, France, octobre 2008.
- [132] The Shimmer Small Wireless Sensor Platform website. <http://shimmer-research.com/wordpress/r-d/platform-features>
- [133] J.-M. Friedt, S. Guinot, Stockage de masse non volatile : un block device pour multimedia card, GNU/LINUX Magazine, Hors-série n° 25, avril/mai 2006.
- [134] J.-M. Friedt, E. Carry, Enregistrement de trames GPS - développement sur microcontrôleur 8051/8052 sous GNU/LINUX, GNU/LINUX Magazine, n° 81, février 2006.
- [135] The basic MMC card access - Retroleum website. Présentation du protocole de communication des SD et MMC en SPI. Available : http://www.retroleum.co.uk/mmc_cards.html
- [136] Spécifications des partitions FAT. Available : <download.microsoft.com/download/1/6/1/161ba512-40e2-4cc9-843a-923143f3456c/fatgen103.doc>
- [137] File allocation table website (Structure d'un support de stockage). available : <http://www.beginningtoseethelight.org/fat16/>
- [138] *The SanDisk Secure Digital Card, Product Manual*, Version 1.9, Document No. 80-13-00169, December 2003 available at <http://www.cs.ucr.edu/~amitra/sdcard/ProdManualSDCardv1.9.pdf>

Travaux de recherche

Bibliographie

- [1] E. P. Capo-Chichi, H. Guyennet, J.-M. Friedt, I. Johnson, and C. Duffy. Design and implementation of a generic hybrid wireless sensor network platform. In *Proceedings of the 33rd IEEE Conference on Local Computer Networks (LCN)*, WLN, pp. 836–840, Montreal, Canada, 2008.
- [2] E. Pamba Capo-chichi and J.-M. Friedt. Design of embedded platform for multimedia application. In *Proceedings of the International Conference on Distributed Frameworks & Applications*, pp. 146–150, Penang, Malaysia, 2008.
- [3] E. Pamba Capo-Chichi, H. Guyennet, and J.-M. Friedt. Ieee 802.15.4 performance on a hierarchical hybrid sensor network platform. In *Proceedings of the Fifth International conference on Networking and Services, ICNS 2009* , pp. 303–308, Valencia, Spain, 2009.
- [4] E. Pamba Capo-Chichi, H. Guyennet, and J.-M. Friedt. K-rle : A new data compression algorithm for wireless sensor network. In *Proceedings of the Third International Conference on Sensor Technologies and Applications, SENSORCOMM 2009*, pp. 502–507, Athens/Glyfada, Greece, 2009.
- [5] E. Pamba Capo-Chichi, D. Martins, H. Guyennet, and V. Felea. A dynamic clustering construction for wireless sensor networks. In *IEEE CTS '09 : Proceedings of the 2009 International Symposium on Collaborative Technologies and Systems*, Poster, page xlv, Washington, DC, USA, 2009.
- [6] E. Pamba Capo-Chichi, J.-M. Friedt, and H. Guyennet. Using Data Compression for Delay Constrained Applications in WSN. *The Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM 2010)*, To appear.

Divers

G. Weisenhorn, E. Pamba Capo-Chichi, J.-M. Friedt - Communications de données et d'images issues de la carte Fox par Bluetooth, GNU/Linux Magazine France 95, Juin 2007.

Résumé

Les récentes avancées dans les divers domaines liés à la micro-électronique, à l'informatique et aux réseaux sans fil ont donné naissance à de nouvelles thématiques de recherche. Les réseaux de capteurs issus de ces nouveaux progrès technologiques constituent un axe de recherche très fertile. En effet, la capacité réduite des noeuds en terme de calcul, de mémoire et d'énergie génère de nombreuses problématiques intéressantes.

Le but de cette thèse est la conception d'une architecture hiérarchique de réseaux de capteurs capable de s'adapter à différents contextes en prenant en compte les contraintes énergétiques et en permettant de fournir des informations riches comme le multimedia à l'utilisateur final.

Nous proposons une architecture hiérarchique avec les différents noeuds qui la composent et les technologies sans fil qui les relient. L'économie d'énergie étant un fil conducteur de notre travail et le module de transmission la principale source d'énergie, nous proposons deux nouveaux algorithmes de compression de données permettant d'optimiser l'utilisation du canal de communication.

Nous présentons également une solution pour le stockage de grandes quantités d'informations sur les noeuds en intégrant le système de fichiers FAT16 sous TinyOS-2.x.

Mots-clés: réseaux de capteurs, économie d'énergie, compression de données, stockage de données.

Abstract

Recent advances in various areas related to micro-electronics, computer science and wireless networks have resulted in the development of new research topics. Sensor networks are one of them. The particularity of this new research direction is the reduced performances of nodes in terms of computation, memory and energy.

The purpose of this thesis is the definition of a new hierarchical architecture of sensor networks usable in different contexts by taking into account the sensors constraints and providing a high quality data such as multimedia to the end-users.

We present our hierarchical architecture with different nodes and the wireless technologies that connect them. Because of the high consumption of data transmission, we have developed two data compression algorithms in order to optimize the use of the channel by reducing data transmitted.

We also present a solution for storing large amount of data on nodes by integrating the file system FAT16 under TinyOS-2.x.

Keywords: sensor networks, energy saving, data compression, data storage.

