



Évaluation de protocoles pour la qualité de service dans les réseaux de capteurs sans fils

Mohamed Bechir Rezgui

► To cite this version:

Mohamed Bechir Rezgui. Évaluation de protocoles pour la qualité de service dans les réseaux de capteurs sans fils. Réseaux et télécommunications [cs.NI]. 2013. dumas-01270767

HAL Id: dumas-01270767

<https://dumas.ccsd.cnrs.fr/dumas-01270767>

Submitted on 8 Feb 2016

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

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

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL RHÔNE-ALPES

CENTRE D'ENSEIGNEMENT DE GRENOBLE

MEMOIRE

présenté par **Mohamed Bechir REZGUI**

en vue d'obtenir

LE DIPLÔME D'INGENIEUR C.N.A.M.

en INFORMATIQUE

**Evaluation de protocoles pour la Qualité de Service
dans les réseaux de capteurs sans fils**

Soutenu le 21 février 2013

JURY

Président : M. Eric Gressier-Soudan (CNAM Paris)

Membres : M. Claude Genier (CNAM Lyon)
M. Jean-Pierre Giraudin (Université de Grenoble)
M. Cyril Labbé (Tuteur, Université Joseph Fourier)
Mme Claudia Roncancio (Grenoble INP)
M. Francis Laffet (Groupe OPEN)

CONSERVATOIRE NATIONAL DES ARTS ET METIERS
CENTRE REGIONAL RHÔNE-ALPES
CENTRE D'ENSEIGNEMENT DE GRENOBLE

MEMOIRE

présenté par **Mohamed Bechir REZGUI**

en vue d'obtenir

LE DIPLÔME D'INGENIEUR C.N.A.M.

en INFORMATIQUE

**Evaluation de protocole pour la Qualité de Service
dans les réseaux de capteurs sans fils**

Soutenu le 21 février 2013

Les travaux relatifs à ce mémoire ont été effectués au sein de l'équipe SIGMA du
Laboratoire Informatique de Grenoble sous la direction de *Cyril LABBE*.

Remerciements

Je tiens à remercier tout d'abord M. Jean-Pierre GIRAUDIN, ex-responsable de la filière informatique au CNAM de Grenoble de m'avoir aidé à trouver ce stage de fin d'étude et pour ses conseils.

Mes remerciements vont également à tout le personnel de l'ancienne équipe administrative du CNAM Grenoble qui m'a assisté tout au long de ma formation en particulier Mme Marie-Ange ARBORE, ainsi que la nouvelle équipe qui a pris la relève.

Je remercie tous les membres du jury, pour avoir accepté de consacrer du temps à la lecture de ce mémoire et me permettre de leur présenter le résultat de ce travail.

Je remercie toute l'équipe SIGMA du Laboratoire d'informatique de Grenoble pour son accueil et sa convivialité.

Mes remerciements se portent aussi vers Mme Claudia RONCANCIO, Professeur à Grenoble INP qui a suivi l'évolution de ce travail.

Je remercie particulièrement M. Cyril LABBE, Maître de conférence à l'université Joseph Fourier, pour s'être investi en tant que tuteur, pour ses conseils précieux, le partage de ses connaissances et pour sa disponibilité tout au long de ce stage.

Je remercie mon épouse Lamia pour ses encouragements, son soutien et surtout sa patience.

Merci à mes sœurs, Ahlem et Intissar et mon frère Atef pour leurs encouragements. Enfin, un grand MERCI à mes parents, Sabiha et Salah, à qui je dédie ce travail, qui malgré les kilomètres qui nous séparent n'ont pas cessé de m'encourager et de me soutenir tout au long de mon parcours.

Sommaire

Liste des figures	v
Liste des tableaux	vii
Acronymes	viii
Introduction générale	1
Contexte général, Problématiques et Objectifs	3
1 Contexte général	3
1.1 Le LIG	3
1.2 L'équipe SIGMA	3
1.3 Le projet ARTEco (Bonus Qualité Recherche Grenoble INP)	4
2 Problématiques et objectifs	5
Chapitre 1 : Etat de l'art	7
1 Les réseaux de capteurs	7
1.1 Architecture des réseaux de capteurs	8
1.2 Caractéristiques des réseaux de capteurs	10
1.3 Les applications à base de capteurs	10
1.3.1 Contraintes	11
2 Gestion des données de capteurs	12
2.1 Les données de capteurs	12
2.2 Gestion des données de capteurs	12
2.2.1 Les systèmes de gestion de bases de données classiques (SGBC)	13
2.2.2 Les systèmes de gestion de flux de données (SGFD)	13
2.3 Les requêtes sur les données des capteurs	15
2.3.1 Les requêtes instantanées	15
2.3.2 Les requêtes historiques	16
2.3.3 Les requêtes continues	16
2.4 SSTREAMWARE : un SGFD générique utilisé dans l'évaluation des requêtes continues	16
3 Qualité de service dans les réseaux de capteurs	19
3.1 Qu'est ce que la qualité de service	19
3.2 Niveaux de service et facteurs impactant la QoS	19
3.2.1 Niveaux de service	19
3.2.2 Facteurs impactant la QoS	20

3.3	Les métriques dans la qualité de service.....	21
3.4	Les mécanismes pour assurer la qualité de service	22
3.4.1	Mécanismes orientés réseau.....	22
3.4.2	Mécanismes orientés données.....	23
4	Consommation et conservation d'énergie	24
4.1	Consommation d'énergie dans les réseaux de capteurs.....	24
4.1.1	Principaux facteurs dans la consommation d'énergie	24
4.1.2	Facteurs induisant une surconsommation d'énergie.....	25
4.2	Conservation d'énergie dans les réseaux de capteurs.....	26
4.2.1	Mécanismes de base pour la conservation d'énergie	27
4.2.2	Techniques orientées réseau	28
4.2.3	Techniques orientées données	29
5	Conclusion	30
Chapitre 2 Le kit Sensinode : Technologie, Matériel et Application		31
1	Sensinode	31
2	La technologie 6LoWPAN	33
2.1	LoWPAN.....	33
2.2	Le standard IEEE 802.15.4	34
2.2.1	La couche physique	34
2.2.2	La couche MAC.....	35
2.2.3	Le transfert des données.....	36
2.3	Le protocole Internet (IP)	36
2.3.1	IPv4	36
2.3.2	IPv6	37
2.4	Fonctionnement de 6LoWPAN.....	39
2.4.1	Description	39
2.4.2	Fragmentation et réassemblage.....	40
2.4.3	Compression d'en-tête	41
2.5	Le protocole de routage RPL	43
2.5.1	Principe de fonctionnement.....	43
2.5.2	Autoconfiguration IP	43
3	Le kit 6LoWPAN de Sensinode.....	45
3.1	Le Nanorouter	45
3.2	Les capteurs Nanosensor N741.....	45

3.2.1	L'unité d'acquisition et de traitement.....	45
3.2.2	L'unité de transmission	46
3.3	NodeView 2.0	48
3.3.1	Présentation	48
3.3.2	Fonctionnement	49
3.3.3	Modification de NodeView 2.0.....	51
4	Conclusion	52
Chapitre 3 : Contributions		53
1	Gestion des données	54
1.1	Programmation des Nanosensors	54
1.1.1	Activation des capteurs	54
1.1.2	Création des messages	55
1.1.3	Envoi des messages	56
1.1.4	Ajout de l'opérateur de sélection.....	59
1.1.5	Flash du microcontrôleur MSP430F5438	61
1.2	Amélioration de NodeView 2.0	62
1.2.1	Ajout de la colonne date de réception.....	62
1.2.2	Ajout de la colonne Modèle du capteur.....	62
1.2.3	Disposition des colonnes	62
1.3	Conclusion	63
2	La qualité de service	64
2.1	Les indices de qualité	64
2.2	Détermination des facteurs impactant la qualité de service	65
2.2.1	Distance entre les Nanosensors et le Nanorouter	65
2.2.2	Les piles utilisées	66
2.2.3	La topologie du réseau	67
2.3	Tolérance aux fautes et qualité de service.....	71
2.3.1	Tolérance aux fautes de mesures.....	71
2.3.2	Tolérance aux pannes matérielles.....	73
2.4	Conclusion	75
3	Conservation d'énergie	77
3.1	Conservation d'énergie au niveau de l'unité d'acquisition et de traitement	77
3.1.1	Eteindre les capteurs non utilisés.....	77
3.1.2	Minimisation de la fréquence d'échantillonnage.....	77

3.1.3	Utilisation des modes basse consommation.....	77
3.2	Conservation de l'énergie au niveau de l'unité de transmission	80
3.2.1	Réduction de la puissance d'émission.....	80
3.2.2	Conservation d'énergie au niveau du module radio	82
3.3	Conclusion	88
Conclusion et perspectives.....		89
1	Conclusion générale	89
2	Perspectives.....	91
2.1	Gestion des données	91
2.2	Consommation d'énergie	91
Annexes		93
1	Configuration et utilisation de IAR Embedded Workbench pour le projet MSP430.....	93
Bibliographie.....		97

Liste des figures

Figure 1 Architecture d'un nœud capteur Source http://www-igm.univ-mlv.fr	7
Figure 2 Réseau de capteurs, liaison directe.....	8
Figure 3 Réseau de capteurs à routage Multi sauts	9
Figure 4 Architecture basée sur les clusters.....	9
Figure 5 Fenêtre glissante, fenêtre sautante, fenêtre bondissante.....	15
Figure 6 SGBD vs SGFD [5].....	15
Figure 7 Architecture orientée services dans SSTREAMWARE [22]	17
Figure 8 Architecture basée sur les clusters.....	27
Figure 9 Internet des Objets [19]	31
Figure 10 Technologie Sensinode basée sur 6LoWPAN [19].....	32
Figure 11 Topologie dans les réseaux LoWPAN	34
Figure 12 Le modèle OSI.....	37
Figure 13 Schéma IPv4	37
Figure 14 Schéma IPv6	38
Figure 15 Structure d'une trame 802.15.4	40
Figure 16 La couche adaptation de 6LoWPAN	40
Figure 17 En-tête d'un fragment 6LoWPAN	41
Figure 18 Trame 802.15.4 avec LoWPAN_IPHC	42
Figure 19 Modes de routage dans 6LoWPAN	42
Figure 20 Structure DAG dans RPL	43
Figure 21 Découverte de voisins dans 6LoWPAN basé sur RPL	44
Figure 22 Activation des capteurs de luminosité et d'accélération	46
Figure 23 Modes de consommation d'énergie de l'unité de transmission	48
Figure 24 Démarrage de NodeView 2.0	49
Figure 25 Connecter le Nanorouter à NodeView 2.0	49
Figure 26 Capteur connecté à NodeView 2.0.....	50
Figure 27 Types de données reçues	50
Figure 28 L'onglet Network analyzer.....	51
Figure 29 L'onglet NanosensorTab.....	51
Figure 30 Les quatre étapes d'une démarche itérative selon la roue de Deming	53
Figure 31 Activation du capteur de luminosité sur le MSP430F5438	55
Figure 32 Le convertisseur Analogique Numérique ADC12	55
Figure 33 ADC12INCH13 Figure 34 Lecture de la valeur luminosité	55
Figure 35 Création des messages	56
Figure 36 Envoi des mesures	57
Figure 37 Cycle de vie d'une mesure.....	58
Figure 38 Opérateur de sélection simple	59
Figure 39 Opérateur de sélection multiple	59
Figure 40 Opérateur de sélection combinée simple	60
Figure 41 Sélection combinée multiple.....	60
Figure 42 Le fichier script.txt	61
Figure 43 Mise à jour du firmware des Nanosensors.....	61

Figure 44 Ajout de la colonne Date de réception.....	62
Figure 45 L'onglet NanosensorTab modifié.....	63
Figure 46 Impact de la distance sur la Qualité de service	65
Figure 47 Impact des piles sur la qualité de service (test 1)	66
Figure 48 Impact des piles sur la qualité de service (test 2)	67
Figure 49 Architecture Multisauts 3 capteurs.....	68
Figure 50 Impact de la topologie du réseau sur la qualité de service (1)	68
Figure 51 Architecture multisauts 10 capteurs	70
Figure 52 Impact de la topologie du réseau sur la qualité de service (2)	70
Figure 53 Mesures de luminosité dans une architecture en étoile utilisant deux capteurs	72
Figure 54 Variation des mesures	72
Figure 55 Mesures de luminosité dans une architecture en étoile utilisant trois capteurs	73
Figure 56 Tolérance aux pannes et couverture réseau (1).....	74
Figure 57 Tolérance aux pannes et couverture réseau (2).....	75
Figure 58 Entrer et sortir du mode LPMx.....	78
Figure 59 Energie en mode LPM	79
Figure 60 Fichier de commande ZNM-CCT	80
Figure 61 ZNM-CCT.....	81
Figure 62 Varier la puissance du signal	82
Figure 63 Durée de vie et Power mode, prise en compte que des paquets de données	84
Figure 64 Wireshark Analyse des trames réseaux.....	85
Figure 65 Durée de vie et Power mode, prise en compte que des échanges réseau	86
Figure 66 Sleep/Wakeup et gigue	87
Figure 67 Création du projet	93
Figure 68 Ajout des fichiers sources.....	94
Figure 69 Ajout des fichiers .h	94
Figure 70 Choix du microcontrôleur.....	95
Figure 71 Format du fichier compilé	95
Figure 72 Compilation du projet MSP430	96
Figure 73 Flasher le microcontrôleur	96

Liste des tableaux

Tableau 1 Caractéristiques du module de transmission [21]	47
Tableau 2 Canaux Radio Fréquences utilisés par RC2400HP	47
Tableau 3 Gestion de la puissance du module RC2400HP	48
Tableau 4 Distance : indices de qualité	66
Tableau 5 Piles et indices de qualité	67
Tableau 6 Topologie et indices de qualité (1)	69
Tableau 7 Changement d'adresses IP	69
Tableau 8 Topologie et indices de qualité (2)	71
Tableau 9 Puissance du signal vs durée de vie et taux de perte	82
Tableau 10 Réseau et data synchrones	85
Tableau 11 Réseau et data asynchrones	86
Tableau 12 Délai de transmission et gigue en utilisant les Power Mode.....	87

Acronymes

6LoWPAN: IPv6 Low power Wireless Personal Area Networks

ABRO: Authoritative Border Router Option

ADC: Analog-to-Digital Converter

AM: Active Mode

ARO: Address Registration Option

BNE: Based Number Evenment

BQR : Bonus Qualité Recherche

CCA: Clear Channel Assessment

CH: Cluster Head

CSMA/CA: Carrier Sense Multiple Access with Collision Avoidance

DAG: Directed Acyclic Graph

DDV : Durée De Vie

DDVM : Durée De Vie Moyenne

DIO: DODAG Information Object

DIS: DODAG Information Solicitation

DODAG: Destination Object Directed Acyclic Graph

ED: Energy Detection

FFD: Full Function Device

GQS: Gateway Query Service

Gw: Gateway

ICMP: Internet Control Message Protocol

IETF: Internet Engineering Task Force

IPSO: IP for Smart Objects

IPv4: Internet Protocol version 4

IPv6: Internet Protocol version 6

LBR: Low power and lossy network Border Router

LPM: Low Power Mode

LQI: Link Quality Indication

LS: Lookup Service

MAC: Media Access Control

MTU: Maximum Transmission Unit

NA: Neighbor Advertissment

NAT: Network Address Translation

NCE: Neighbor Cache Entry

ND: Neighbor Discovery

OSI: Open Systems Interconnection

PAN: Personnal Area Network

PM: Power Mode

PQS: Proxy Query Service

PSDU: Physical layer Service Data Unit

QdS : Qualité de Service

QoS : Quality of Service

RA: Router Annoucement

RAM: Random Access Memory

RFC: Request For Comment

RFD: Reduced Function Device

RISC: Reduced instruction set computing

RPL: Routing Protocol for Low power and Lossy Network

RS: Router Sollicitation

SGBC: Systèmes de Gestion de Bases de Données Capteurs

SGBD : Système de Gestion de Base de Données

SGFD : Systèmes de Gestion de Flux de Données

SI : Système d'Informations

SoC: System on Chip

SQS: Sensor Query Service

TCP: Transmission Control Protocol

TDMA: Time Division Multiple Access

UDP: User Datagram Protocol

USB: Universal Serial Bus

Introduction générale

Au cours des dernières années, nous avons assisté d'une part à un développement très rapide et continu des réseaux de télécommunications notamment les réseaux sans fils et d'autre part à des progrès spectaculaires de la technologie, notamment la miniaturisation des composants électroniques et la puissance croissante de calcul. Tous ces éléments ont permis une évolution des équipements informatiques qui sont devenus de plus en plus petits, puissants et autonomes.

Les capteurs, sont devenus omniprésents. Ils forment une «peau virtuelle» du monde réel et nous informent des événements physiques se déroulant autour de nous. Les réseaux de capteurs sans fils constituent l'un des domaines de recherche les plus abordés. La valeur ajoutée de ce type de réseaux fait partie de notre quotidien puisque les domaines d'applications sont multiples et variés. Parmi ces applications, nous pouvons citer [1]:

- les applications environnementales : par exemple la mesure de la température ou de la luminosité dans un endroit ;
- les applications militaires : par exemple la localisation d'armes ou de véhicules ;
- les applications médicales : par exemple la surveillance du rythme cardiaque d'une personne ;
- les applications domotiques avec les projets de maisons intelligentes ;
- les applications industrielles par exemple le contrôle des procédés dans une chaîne de fabrication.

Pour toutes ces applications, les capteurs ont toujours le rôle d'effectuer des mesures et générer des données qui sont par la suite traitées par ces applications. Les données peuvent être traitées de deux façons différentes. Soit en utilisant le système de gestion de base de données classique où les données sont stockées et des requêtes sont émises par les applications pour récupérer celles qui l'intéressent. Soit le traitement est réalisé par un système de gestion de flux de données où les données arrivent en continu.

Une telle généralisation de l'utilisation des capteurs a pour conséquence leur introduction dans de nombreuses applications où la qualité de service et la sûreté de fonctionnement sont des enjeux importants sans pour autant s'agir de systèmes critiques. En outre, l'installation des capteurs dans des environnements parfois hostiles ou difficiles d'accès et leur dépendance vis-à-vis d'une source d'énergie limitée en font des équipements relativement vulnérables. Enfin, la complexité croissante des systèmes et des applications utilisant des capteurs et des dispositifs accroît leur propension à la défaillance, ce qui va à l'encontre de la qualité de service et de la sûreté de fonctionnement.

Les différentes applications basées sur les réseaux de capteurs ont chacune des attentes différentes en termes de performances, ce qui implique des contraintes très différentes, parfois complexes, en termes de réactivité (latence tolérée), d'intensité de trafic, de sécurité, de densité, etc. Le dénominateur commun entre toutes ces applications est la nécessité d'avoir une bonne qualité de service tout en minimisant les coûts énergétiques et maximisant la durée de vie du réseau.

Dans la plupart des cas, les capteurs sont équipés de batteries, ce qui veut dire une quantité d'énergie limitée. Dans certains cas où l'environnement le permet, une certaine quantité d'énergie peut être récupérée grâce à des modules spécifiques qui se basent sur la lumière ou la chaleur dégagée. Cela reste insuffisant pour prolonger la durée de vie des capteurs d'une manière considérable. L'énergie est donc un enjeu très important dans le progrès et l'évolution de ces réseaux.

De nombreux travaux ont étudié la problématique de la tolérance aux pannes dans les applications à réseaux de capteurs selon diverses approches : la redondance matérielle et temporelle, la reconstruction topologique du réseau, le routage multi-chemins, etc. Ces approches se distinguent par de nombreux facteurs : la couche protocolaire d'implémentation, les types de pannes qu'elles traitent (pannes franches, de mesures, de transmission ou bugs logiciels), les types d'applications ciblées, le niveau d'autonomie, le coût d'implémentation (en termes de ressources matérielles, d'énergie, de surcharge en communication, d'occupation du processeur, de l'espace mémoire, etc.). Ces approches se limitent à un niveau donné du système de capteurs : matériel, réseau de communication, application, etc. Le travail qui m'a été demandé dans ce stage consiste à prendre en main un nouveau matériel de Sensinode utilisant la technologie 6LoWPAN. Il s'agit de gérer les données, étudier son comportement par rapport au support de la qualité de service notamment la tolérance aux pannes et optimiser la consommation d'énergie.

Ce travail a été réalisé dans le cadre d'un stage de neuf mois que j'ai effectué au sein de l'équipe SIGMA¹ du LIG² en vue d'obtenir le diplôme d'ingénieur CNAM en informatique. Il fait partie du projet ARTEco (BQR Grenoble INP) que nous allons présenter dans la partie contexte général et concerne la qualité de service dans les réseaux de capteurs, avec un accent particulier sur la tolérance aux fautes et le souci d'une réduction de la consommation énergétique. Un kit de capteurs de Sensinode était à ma disposition pour effectuer les différents travaux et expérimentations.

Dans ce mémoire, nous commençons par présenter le contexte général du projet ainsi que les problématiques posées et les objectifs à atteindre. Nous présenterons dans le chapitre 1 un état de l'art sur les réseaux de capteurs, la gestion des données, la qualité de service et l'énergie. Vient ensuite le chapitre 2 dans lequel, nous présenterons le matériel utilisé ainsi que les différents protocoles, technologies et mécanismes sur lesquels est basé notre réseau de capteurs Sensinode. Dans le chapitre 3, nous détaillerons les contributions apportées au projet, il contiendra les tests et expérimentations réalisés et nous finirons par une conclusion générale.

¹ Systèmes d'information – Ingénierie et Modélisation Adaptable <http://sigma.imag.fr>

² Laboratoire d'informatique de Grenoble <http://liglab.fr>

Contexte général, Problématiques et Objectifs

La dernière unité d'étude (UE) dans le parcours de la formation « Ingénieur Système, Réseaux et Multimédia » du CNAM avant de commencer le stage de fin d'étude était l'épreuve TEST (Travail d'Etude et de Synthèse Technique). Le travail que j'ai effectué s'intitulait « Qualité de service et énergie dans l'interrogation de capteurs hétérogènes » et était encadré par Cyril LABBE de l'équipe SIGMA du LIG. Ce stage vient donc comme une suite logique à l'épreuve TEST. Dans cette partie, nous allons décrire le contexte général du projet, les problématiques posées et enfin les objectifs à atteindre.

1 Contexte général

Dans cette partie, nous allons présenter l'environnement où j'ai effectué mon stage ainsi que le projet auquel sont liés mes travaux. En premier je présente le LIG, ensuite l'équipe SIGMA et enfin le projet ARTEco.

1.1 Le LIG



Le LIG [2] a été créé en janvier 2000 au terme d'une restructuration de plusieurs laboratoires grenoblois. Il rassemble près de 500 chercheurs, enseignants, doctorants et techniciens répartis en 24 équipes de recherches. Ses partenaires académiques sont le CNRS (Centre National de la Recherche Scientifique), Grenoble INP (Institut National Polytechnique), l'INRIA (Institut National de Recherche en Informatique et Automatique), les universités Joseph Fourier et Pierre Mendès France.

1.2 L'équipe SIGMA



L'équipe SIGMA [3] réalise des recherches sur les systèmes d'information aussi bien aux niveaux organisationnel que opérationnel. Ces recherches sont centrées sur la formalisation, la conception et les infrastructures des systèmes d'Information.

Un volet important des recherches de l'équipe SIGMA est centré sur les modèles, leur évaluation et leur transformation pour favoriser principalement quatre aspects : précision, flexibilité, traçabilité et réutilisation. Les travaux portent sur des thèmes fondamentaux comme de nouveaux modèles ou métamodèles pour certains orientés processus métier ou intrication de contextes collectifs ou personnalisés. Ces aspects fondamentaux sont en permanence confrontés à des recherches appliquées avec quatre domaines privilégiés : Santé, Produit, Transport et Enseignement, où l'impact de l'informatique ubiquitaire est aussi à étudier.

En particulier, les systèmes d'information de santé nécessitent de trouver des modèles permettant de les rendre ubiquitaires, coopérants et adaptables au contexte d'utilisation. L'équipe Sigma mène des recherches notamment sur des modèles pour les interfaces utilisateurs en médecine, l'adaptation contextuelle et la dimension ubiquitaire des SI de santé.

Concernant les aspects infrastructures des systèmes d'information ubiquitaires et grande échelle, l'accent est mis sur la gestion cohérente et performante des informations qui prennent en compte des données sur des mobiles et issues de capteurs. Le projet ARTEco se situe dans cet axe de recherche. Nous allons le présenter dans le prochain paragraphe.

1.3 Le projet ARTEco (Bonus Qualité Recherche Grenoble INP)

ARTEco est l'abréviation de : **A**pplications à base de **R**éseaux de **C**apteurs **T**olérantes aux **F**autes & **E**nergétiquement **E**conomiques. Ce projet est porté par l'équipe SIGMA du LIG et le LCIS (Laboratoire de Conception et d'Intégration des Systèmes) de Valence.

ARTEco consiste en la détection de fautes dans les systèmes de capteurs ainsi que la tolérance aux fautes de traitement de l'information issue de ces capteurs. Ce projet s'inscrit complètement dans les thèmes de recherche de l'équipe, où la prise en compte des capteurs est abordée dans les thèmes « Infrastructure des SI » et « SI santé » notamment au travers de l'habitat intelligent que l'équipe commence à aborder (à la fois pour la santé ou la vie quotidienne). Les résultats escomptés du projet ARTEco vont composer une brique importante pour cette nouvelle recherche. Les travaux menés s'insèrent dans le cadre de recherches plus larges qui visent à faciliter l'utilisation des capteurs dans les grands systèmes d'information.

La qualité de service constitue l'axe central du projet. En effet, pour nombre d'applications, il est nécessaire de pouvoir garantir une certaine « qualité » du monitoring et donc des informations extraites via le réseau de capteurs. Ces critères de qualité peuvent être de différents types, par exemple :

- disposer d'informations issues d'un pourcentage minimum de capteurs à chaque instant ou sur un intervalle de temps ou bien disposer d'au moins une donnée issue de capteurs dans une certaine zone. Ainsi, lors d'un transport de bouteilles de vin, on veut s'assurer que la température et l'humidité restent stables. Pour cela, il faut avoir des données de capteurs issues de plusieurs endroits du stock de vin et à des intervalles de temps réguliers ;
- disposer d'au moins une donnée ayant une certaine précision (au moins un capteur de température par zone de stockage de vin doit fournir une précision d'un demi degré).

Pour offrir des telles garanties de qualité aux applications, il est nécessaire de répercuter les besoins de qualité sur la chaîne d'interrogation. Cela inclut le niveau capteur individuel qui relève des mesures, le réseau de capteurs, la collecte des données et l'évaluation des requêtes continues destinées à calculer l'information effectivement requise par les applications. Il est important alors de garantir la qualité de fonctionnement à divers niveaux tout en préservant les ressources cruciales des capteurs (bande passante, énergie, etc.).

La seconde motivation d'ARTEco concerne la réduction de la consommation énergétique tout au long de la chaîne de traitement des mesures relevées par les capteurs. En effet, ce point est particulièrement important dans les systèmes basés sur ce genre de dispositifs. C'est le point de départ pour arriver à assurer une continuité de service et permettre aux applications d'offrir aux utilisateurs une meilleure qualité de fonctionnement. Une première solution générique a été proposée dans le cadre de ce projet qui permet l'adaptation permanente de l'évaluation des requêtes continues en fonction de l'état des capteurs sous-jacents et des besoins en qualité de service des applications (précision, fréquence des réponses, préservation de l'énergie, etc.). Cette solution permet l'économie des ressources aux différentes étapes de l'évaluation des requêtes continues : sur les capteurs, stations de base, serveurs en ne traitant que les données strictement nécessaires aux requêtes en cours d'exécution. Nous allons travailler pour apporter d'autres possibilités de conservation d'énergie au niveau de la source des données.

2 Problématiques et objectifs

Tout d'abord, il faut savoir que le matériel utilisé pendant le stage venait d'être acquis par l'équipe SIGMA, donc la première problématique était la prise en main théorique et pratique de ce matériel. Il s'agit d'un kit de dix capteurs plus un routeur, le tout fonctionnant en IPv6.

La première étape de ce travail était de se familiariser avec les réseaux de capteurs, la gestion des données, les modèles de fautes et la mesure de la qualité de service.

Ensuite, pouvoir appliquer les notions acquises sur notre matériel et avoir la possibilité de l'exploiter afin de l'intégrer dans le projet ARTEco. En d'autres mots, arriver à gérer les données, définir des paramètres de qualité de service et enfin augmenter la durée de vie des capteurs sachant que ces derniers sont alimentés par des piles. Les objectifs sont donc d'arriver à :

- Programmer les capteurs pour les mesures, l'échantillonnage et l'envoi des paquets ;
- Adapter l'application fournie par Sensinode à nos besoins en terme de gestion des données et communication avec les capteurs ;
- Proposer des solutions pour le support de la qualité de service ;
- Etudier le comportement des capteurs en termes de consommation d'énergie et proposer des idées et solutions afin de réduire au maximum cette consommation.

Le premier chapitre de ce travail est un état de l'art sur les réseaux de capteurs et les différentes problématiques et solutions proposées liées à nos préoccupations afin d'atteindre les objectifs fixés. Dans ce chapitre, nous allons présenter dans la première section les réseaux de capteurs et les applications liées. Ensuite nous présenterons dans la deuxième section l'aspect gestion des données, en effet selon l'application utilisée, des systèmes sont mieux adaptés que d'autres. La section 3 contiendra une présentation de la qualité de service dans les réseaux de capteurs et les différentes métriques utilisées pour la qualifier et l'améliorer. Nous remarquerons que la consommation d'énergie est une de ces métriques si ce n'est pas la plus importante c'est pour cela d'ailleurs que nous consacrerons la section 4 pour présenter les acteurs jouant dans la consommation et ensuite les propositions faites afin de conserver l'énergie et rallonger la durée de vie.

Chapitre 1 : Etat de l'art

Dans cette partie, nous allons présenter dans la section 1 les réseaux de capteurs, les différentes architectures que nous pouvons rencontrer, leurs caractéristiques ainsi que quelques types d'applications à base de réseaux de capteurs. Ensuite dans la section 2, nous nous concentrerons sur la gestion des données issues de ces dispositifs. Dans la section 3, nous décrirons les aspects liés à la qualité de service et enfin dans la section 4 nous présenterons les sources de consommation de l'énergie ainsi que quelques techniques pour la conserver qu'on essayera d'appliquer sur notre matériel Sensinode.

1 Les réseaux de capteurs

Un nœud capteur est un système qui permet de mesurer et transformer une grandeur physique en donnée exploitable. Il permet donc de convertir en valeur numérique une donnée environnementale mesurée. Un capteur peut regrouper plusieurs sous-systèmes responsables chacun d'une ou plusieurs tâches. Il permet l'acquisition des données, leurs traitements et leurs transmissions. Un capteur sans fils est également équipé d'une source d'énergie [4].

Un nœud capteur est représenté par quatre unités (cf. figure 1) :

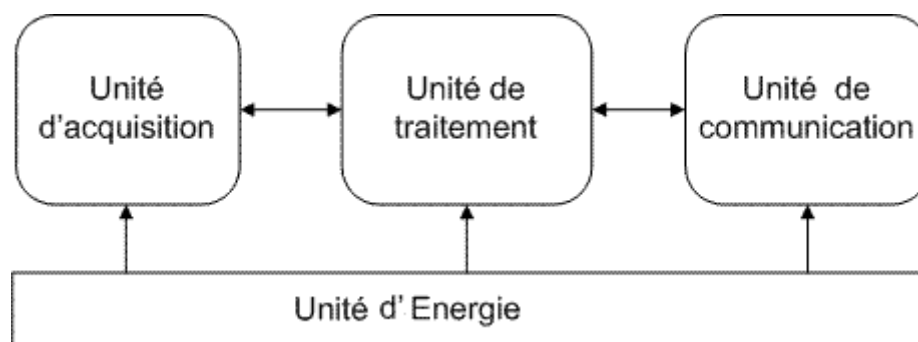


Figure 1 Architecture d'un nœud capteur Source <http://www-igm.univ-mlv.fr>

- Unité d'acquisition : elle est responsable de la capture des événements du milieu physique et de la conversion du signal analogique vers un signal numérique ;
- Unité de traitement : les événements captés sont transformés en données numériques et sont traités par le processeur ;
- Unité de communication : elle est composée d'un module radio (émetteur/récepteur) permettant l'échange des données entre les différents nœuds ou entre les capteurs et la station de base ;
- Unité d'énergie : c'est la batterie qui dans la plupart des cas n'est ni remplaçable ni rechargeable.

Comme pour les équipements informatiques, un système d'exploitation est mis en place afin de permettre à toutes les unités d'accomplir leurs tâches. Les systèmes les plus connus sont TynOS et Contiki.

1.1 Architecture des réseaux de capteurs

Un réseau de capteurs est composé de plusieurs capteurs mis en réseau avec une station de base dite « Puits » qui collecte les informations transmises par ces derniers.

Ce genre de réseau a connu à travers le temps une évolution de point de vue architecture. Cette évolution concerne la façon avec laquelle les capteurs sont organisés au sein du réseau et la manière qu'ils utilisent pour communiquer entre eux ou avec la station de base. On distingue trois types d'architecture :

- Les réseaux à liaison directe [4] :

Dans ces premiers réseaux, les capteurs communiquaient directement avec le « puits ». Cette architecture est illustrée dans la figure 2.

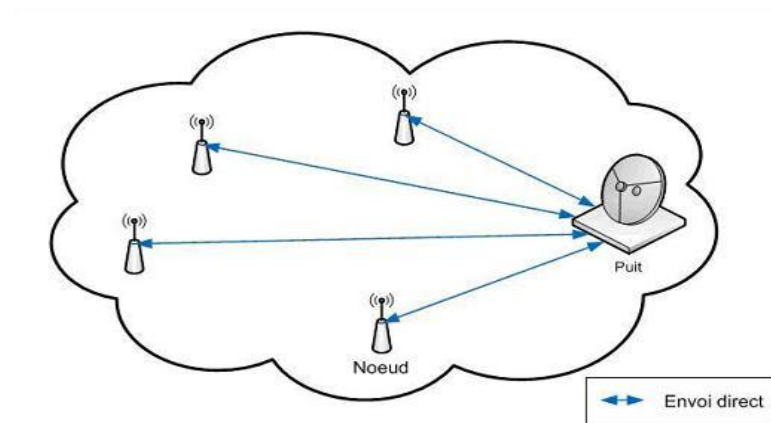


Figure 2 Réseau de capteurs, liaison directe

Ce sont les premières architectures qui ont été élaborées, le routage dans ce type d'architecture ne présentait pas un problème puisque tous les nœuds du réseau savaient où envoyer l'information.

- Réseaux multisautes :

Dans ce cas, les nœuds capteurs sont interconnectés entre eux et l'acheminement de l'information jusqu'à la station de base peut se faire de deux façons. Les données sont acheminées directement vers le nœud « puits » si le capteur est à portée de ce dernier. Dans le cas où les données proviennent d'un capteur loin du nœud puits, celles-ci vont passer par les autres capteurs intermédiaires jusqu'à arriver à la station de traitement (cf. figure 3).

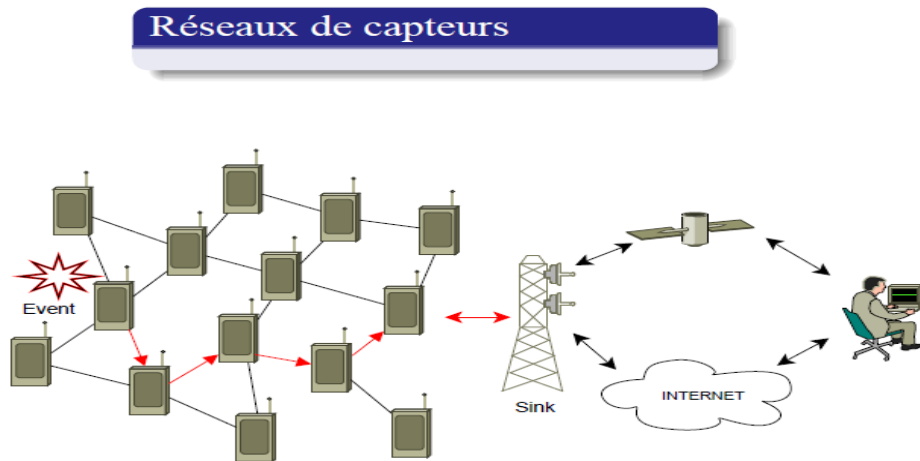


Figure 3 Réseau de capteurs à routage Multi sauts

Les données collectées par les stations de bases sont ensuite acheminées vers des machines puissantes qui auront la tâche de les analyser et les exploiter [5].

- Les réseaux à base de clusters [6]:

Dans ce type de réseaux, les deux premières techniques sont combinées. En effet, il s'agit de former des petits réseaux (cluster) au sein d'un seul réseau. Pour chaque cluster, on trouve un Clusterhead (CH) et les autres capteurs. Ces derniers ne vont échanger les données qu'avec le CH, cette partie constitue le réseau à liaison directe. Ensuite, les CH vont utiliser la technique multisauts pour acheminer l'information jusqu'à la station de base (voir figure 4).

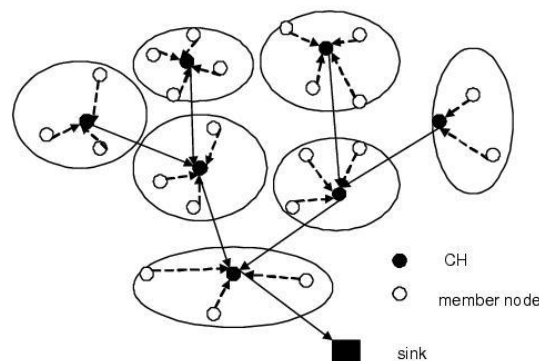


Figure 4 Architecture basée sur les clusters

Dans un seul réseau, on peut rencontrer différents types de capteurs qui ont la charge de récolter des informations différentes selon le domaine d'intérêt et les événements environnementaux que l'on souhaite surveiller. On parle d'hétérogénéité de capteurs.

1.2 Caractéristiques des réseaux de capteurs

Une des caractéristiques principales dans les réseaux de capteurs est le nombre de capteurs déployés. Ce nombre peut aller de quelques dizaines à quelques centaines voire à quelques milliers de capteurs dans un seul réseau. Ceci est dans le but de garantir une couverture totale de la zone à surveiller. Cette densité importante peut entraîner un changement de topologie fréquent. Ce changement est dû aussi aux capteurs mêmes qui peuvent rencontrer des défaillances ou des dysfonctionnements liés principalement à leurs sources d'énergie ou tout simplement à des défauts de fabrication [1, 5].

La possible mobilité des nœuds est un facteur qui conduit au changement de la topologie dans un réseau de capteurs. En effet, dans certains cas, les capteurs peuvent être attachés à des dispositifs mobiles qui peuvent se déplacer librement dans la nature.

Un autre facteur qui peut conduire au changement de la topologie d'un réseau est tout simplement l'ajout de nouveaux capteurs.

Comme autres caractéristiques principales, nous pouvons citer :

- Les ressources limitées : les capacités des capteurs sont limitées au niveau du calcul, du stockage et de la communication. Ceci est dû à leur taille qui peut atteindre quelques millimètres cube. Les ressources en énergie restent assez limitées et présentent un point de faiblesse [1].
- L'imprécision des mesures : dans un réseau de capteurs, le support souvent utilisé pour la transmission des données est le média sans fils. Les paquets transmis sont donc exposés aux obstacles et subissent des pertes, des collisions et des interférences [1].

1.3 Les applications à base de capteurs

On distingue cinq grandes familles d'applications [7] :

- Les applications militaires : leurs objectifs incluent la surveillance des forces amies, des équipements, des munitions et du champ de bataille, la détection des attaques (biologiques, chimiques, etc.), la reconnaissance des forces ennemies, l'évaluation des dommages, etc. Un exemple d'application militaire est VigilNet [24]. C'est un réseau de surveillance à grande échelle conçu pour le pistage de cibles dans des environnements difficiles.
- Les applications pour l'environnement : comme par exemple la détection de feux dans les forêts, la détection d'inondation, l'agriculture de précision, le recensement et la surveillance d'animaux, etc. Le projet "Great Duck Island" [25] est un exemple d'application pour l'environnement. C'est le résultat d'une collaboration entre "College of Atlantic" et "Intel Research Laboratory" à Berkeley pour étudier la distribution et l'affluence des oiseaux de mer à Great Duck Island, Maine.
- Les applications médicales : comme celles permettant la télésurveillance des patients et de leurs fonctions vitales, le repérage des patients, des médecins et du matériel dans un hôpital, etc. Le projet "Appartement intelligent pour l'assistance à la personne" est un exemple d'application médicale. Ce projet, est mené par les équipes Trio, Madynes, Maia et Cortex du Loria [7] et vise des applications relatives à

la domotique et à l'assistance à la personne dans des cadres aussi bien professionnels que médicaux ou privés (exemple : l'aide cognitive pour des professions où la surcharge en information est grande, le maintien à domicile des personnes âgées ou dépendantes). Ce projet fait partie aussi de la quatrième famille, celle des applications pour la maison.

- Les applications industrielles : la supervision, la localisation et l'inventaire des équipements industriels sont des exemples d'applications industrielles. Par exemple, la société fireflies-rtls propose une solution de localisation de biens en temps réel destinée aux industriels.

Selon le type d'applications et son environnement de déploiement, nous pouvons être confrontés à des contraintes qui auront une influence directe sur la qualité de service.

1.3.1 Contraintes

Plusieurs contraintes sont liées aux réseaux de capteurs sans fils. Les principales sont :

- Le support utilisé : c'est le média sans fils qui par nature n'est pas fiable et son état varie dans le temps et est fortement lié à l'environnement de déploiement ;
- La densité du réseau : comme nous l'avons déjà mentionné dans le paragraphe « caractéristiques des réseaux de capteurs », ces derniers sont déployés dans certains cas par milliers. Ceci doit être pris en compte dans le développement des algorithmes liés à la gestion des données et ceux liés à la configuration du réseau ;
- Le matériel : cette contrainte est liée en partie à la densité du réseau. En d'autres mots, le coût de production des capteurs ne doit pas être trop élevé puisque ces derniers peuvent être déployés par milliers. Donc, optimiser le coût de production revient à optimiser le coût du matériel et par conséquent nous aurons des capteurs à faible capacité de mémoire, de puissance de calcul et de traitement ;
- L'énergie : c'est la contrainte la plus forte dans les réseaux de capteurs sans fils. En effet, dans la plupart des cas, la source d'énergie utilisée sont des batteries dont la durée de vie est limitée. Cette contrainte impacte directement la qualité de service c'est pour cela, multiples sont les travaux proposés pour conserver cette ressource. Nous détaillerons quelques mécanismes dans les parties suivantes de cet état de l'art.

Toutes ces contraintes rendent difficile la garantie d'une qualité de service optimale surtout quand il s'agit d'applications critiques où la tolérance aux fautes et l'imprécision de mesures par exemple ne sont pas acceptées. A ces contraintes, s'ajoute l'aspect gestion de données provenant de ces capteurs.

2 Gestion des données de capteurs

Dans une même zone, nous pouvons rencontrer différents types de capteurs provenant de différents fabricants, responsables d'assurer plusieurs tâches et de fournir plusieurs sortes d'informations, d'où l'hétérogénéité de ces derniers. La gestion des données doit alors prendre en compte cet aspect. Les fabricants s'intéressent aussi à la partie applicative et exploitation des données issues des capteurs, ils proposent donc des solutions logicielles propres à leurs capteurs.

2.1 Les données de capteurs

Les mesures relevées par les capteurs sont de différents types et natures variées. Cela dépend de l'environnement où ils sont déployés, des types d'informations qu'ils collectent et du domaine d'application [1].

Dans une zone d'intérêt, on peut rencontrer différents types de capteurs qui peuvent avoir la charge de détecter les événements de type présence de personne, sonore ou visuel, la température ou la pression d'un endroit, etc.

L'hétérogénéité concerne donc les trois points suivants :

- Le type du capteur et son domaine d'application.
- Les fabricants.
- Les logiciels d'interprétation des données.

Les capteurs, après leur mise en place, deviennent responsables des tâches qui leurs sont confiées : la surveillance de l'environnement, le traitement des informations relevées et enfin leur transmission.

Ils sont donc en permanence en train d'émettre des données. Les données sont émises en continu et en flux. Selon le domaine d'application et les données recherchées, nous pouvons les programmer de telle sorte que l'information soit envoyée périodiquement, par exemple toutes les 2 minutes ou à l'occurrence d'un événement particulier, par exemple après dépassement d'un seuil.

Dans la plupart des cas, nous associons à ces données des valeurs supplémentaires (méta-données) comme par exemple l'identificateur du capteur, sa position ou aussi une estampille de temps [1]. On appelle donc données de capteurs l'ensemble des valeurs émises par ces derniers. Le but est donc de pouvoir extraire de ces données des informations pertinentes. Il faudra gérer ces données le mieux possible afin de pouvoir les exploiter.

Dans le paragraphe suivant, nous allons aborder les différents systèmes de gestion des données issues de capteurs.

2.2 Gestion des données de capteurs

Les systèmes de gestion des données de capteurs ont fait l'objet de plusieurs recherches. On cherche toujours à optimiser le traitement de ces données en tenant compte des contraintes que connaît ce genre de réseau. Historiquement, chaque application utilisait des capteurs spécifiques avec des méthodes propres à elle même pour interroger ces derniers mais ceci devient impossible avec la multiplication des domaines d'applications.

La première approche qui a été mise en place est une approche orientée base de données, on parle de systèmes de gestion de bases de données classiques (SGBC).

2.2.1 Les systèmes de gestion de bases de données classiques (SGBC)

Le principe utilisé dans les bases de données capteurs est similaire aux systèmes de gestion de bases de données traditionnels. En effet, les capteurs vont être considérés comme des sources de données, par conséquent le réseau de capteurs formera une base de données distribuée.

« Une base de données capteurs est un ensemble conceptuel de données de capteurs qui représentent les mesures faites sur l'environnement physique. Les données ne sont pas stockées dans cette base. Elles sont instantanées, transitoires, et fréquemment variables. » L. GÜRGEN. [1]

L'interrogation de la base de données se fait à partir de la station de base à travers l'envoi de requêtes sur le réseau. Avec ce type de mécanisme, une fois la requête reçue par le capteur, elle est évaluée localement. La station de base ne recevra des réponses qu'en provenance des capteurs concernés par la requête envoyée.

Les avantages des bases de données capteurs sont les suivants :

- Le nombre de paquets circulants sur le réseau est diminué puisque le traitement se fait en local ;
- Pas de saturation au niveau de la station de base. En effet cette dernière ne recevra que les données qui seront générées suite aux requêtes qu'elle a envoyé ;
- Pas de délai pour les résultats puisque l'évaluation des requêtes se fait au niveau des capteurs ;
- Les capteurs consomment moins d'énergie en utilisant leurs ressources de stockage et de traitement que s'ils utilisaient leurs unités de transmission.

Malgré tous ces avantages, ce système de traitement de données de capteurs reste insuffisant parce que les capteurs ne cessent d'évoluer ainsi que leurs caractéristiques. Les applications utilisant les capteurs comme source de données sont de plus en plus gourmandes en termes de temps de traitement des requêtes et de temps de réponse.

Les chercheurs se sont donc lancés dans d'autres travaux concernant la gestion des données de capteurs en prenant en compte leurs spécificités, les données collectées, les contraintes rencontrées dans ce type de réseaux et surtout l'hétérogénéité.

Ceci a conduit à un autre mode de gestion des données de capteurs, le système de gestion de flux de données.

2.2.2 Les systèmes de gestion de flux de données (SGFD)

2.2.2.1 Flux de données

Le concept flux de données est né suite au développement très rapide des applications qui génèrent une quantité très importante de données [8]. La masse de données générée combinée au besoin de les traiter en temps-réel a conduit au fait qu'on ne peut plus utiliser les systèmes traditionnels de gestion des données comme par exemple les systèmes de gestion de bases de données classiques (SGBC).

Un flux de données est une séquence infinie de données, générées de façon continue. Cette séquence est composée de n-uplets, non-bornée et ordonnée par une estampille de temps [1, 11].

L'aspect infinité des données doit être pris en compte lors de l'évaluation des requêtes continues. En effet, certaines opérations doivent avoir en entrée un ensemble fini de données sinon l'opération est bloquée. Afin de remédier à ce genre de problème, il faudra, à partir du flux infini, créer un ensemble fini de données. Cet ensemble fini est appelé « fenêtre ». Ce point sera expliqué dans le prochain paragraphe.

Une définition d'un flux de données a été donnée par Golab et Özsu dans leur article [9] : *« Un flux de données est une séquence d'items continue, ordonnée (implicitement par temps d'arrivée dans le Système de Gestion de Flux de Données, ou explicitement par timestamp de production à la source), arrivant en temps réel. Il est impossible de contrôler l'ordre dans lequel arrivent les items, et il est impossible de stocker localement un flux dans sa globalité. Les requêtes sur un flux de données sont actives continuellement sur une période de temps et retournent incrémentalement de nouveaux résultats lorsque des nouvelles données arrivent. Ces requêtes sont dites persistantes, continues et de longue durée d'exécution ».*

Les réseaux de capteurs se situent dans ce contexte. En effet, les capteurs sont amenés à générer des données de façon continue et en flux.

2.2.2.2 Les fenêtres

Les fenêtres ont été créées pour faire face aux opérateurs dits bloquants qui ne peuvent pas s'exécuter en présence d'un flux infini de données. Parmi les opérateurs bloquants, nous pouvons citer : somme, moyenne, max, jointure, etc. Il existe deux types de fenêtres dans les SGFD :

- Les fenêtres temporelles

Elles sont basées sur le temps. La largeur est définie par une durée qu'on spécifie dans la requête. Un exemple de requête basée sur des fenêtres temporelles : *quelle est la moyenne des mesures de température dans la pièce A depuis 5 minutes ?*

- Les fenêtres BNE

Les fenêtres BNE (Based Number Evenment) sont basées sur le nombre d'événements. Un exemple de requête utilisant les fenêtres BNE : *Quelle est la moyenne des cinq dernières mesures de température dans la pièce A ?*

Un autre élément à prendre en compte lorsqu'on souhaite définir des fenêtres dans un SGFD est le décalage entre celles-ci. Classiquement il existe trois types de décalage [10] (cf. figure 5) :

- Fenêtres glissantes (sliding window) : le pas de décalage est inférieur à la taille de la fenêtre.
- Fenêtres sautantes (jumping window) : le pas de décalage est égal à la taille de la fenêtre.
- Fenêtres bondissantes (hopping window) : le pas de décalage est supérieur à la taille de la fenêtre.

La Figure 5 illustre les trois types de décalage.



Figure 5 Fenêtre glissante, fenêtre sautante, fenêtre bondissante

2.2.2.3 Traitement de flux de données

Dans un SGFD, les données sont transitoires tandis que les requêtes sont persistantes alors que dans un SGBD les données sont persistantes et les requêtes sont instantanées. A ce stade, nous pouvons dire qu'il y a au moins un avantage des SGFD par rapport aux SGBD : dans un SGFD, les requêtes sont évaluées continuellement sur les nouvelles données arrivant dans le système alors que dans les SGBD, c'est au système d'aller chercher les nouvelles informations.

Dans la Figure 4, nous pouvons voir une illustration des deux systèmes :

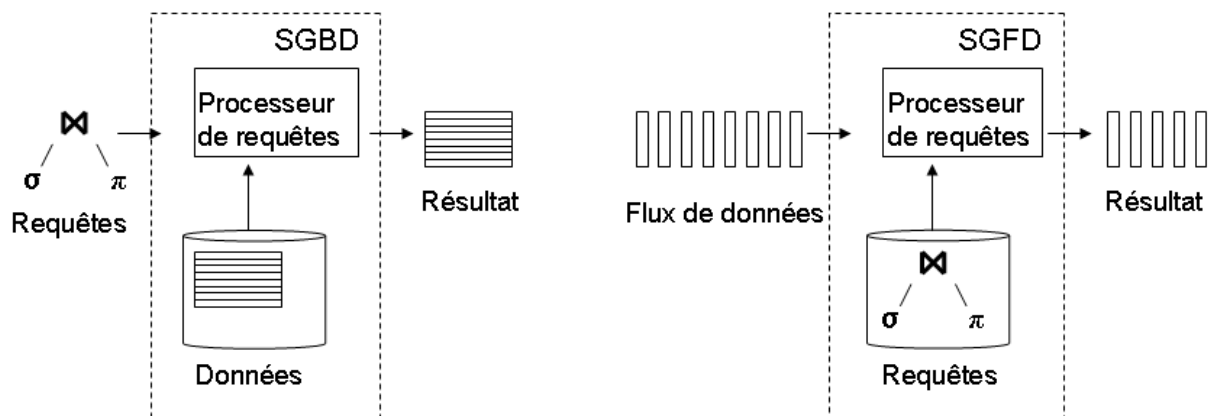


Figure 6 SGBD vs SGFD [5]

Dans les deux cas, SGBD ou SGFD, nous avons besoin d'envoyer des requêtes afin de pouvoir extraire les données qui nous intéressent. Dans le paragraphe suivant, nous allons présenter les différents types de requêtes dans l'interrogation des réseaux de capteurs.

2.3 Les requêtes sur les données des capteurs

On distingue deux types de requêtes dans l'interrogation des capteurs, les requêtes instantanées et les requêtes continues.

2.3.1 Les requêtes instantanées

Comme son nom l'indique, ce type de requêtes est évalué une seule fois à un instant t sur l'ensemble des données du système disponibles à cet instant. Cela nous permet de connaître l'état du système au présent comme au passé, cela dépend de t . Ce type de requêtes est utilisé dans le cas où nous utilisons un système de gestion de bases de données c'est-à-dire que les données sont stockées et peuvent être consultées à tout moment. Cela correspond aux requêtes classiques des SGBD.

Un exemple de ces requêtes :

Quelle est la température de la pièce X en ce moment ?

Un dérivé des requêtes instantanées est la requête historique.

2.3.2 Les requêtes historiques

Généralement, les capteurs ne conservent pas leurs données, car une fois émises on ne leur demande pas de les conserver, ceci est dû à la faible capacité de stockage de ces derniers. L'évaluation des requêtes historiques peut se faire seulement dans le cas où on est en présence d'un système qui permet la persistance des données (SGBD).

Un exemple de ces requêtes :

Quelle est la température de la pièce X hier à 23 :00 ?

2.3.3 Les requêtes continues

Ce type de requête est utilisé dans le cas où les données sont émises en continu, en d'autres mots quand on est en présence de flux de données. Le flux de données s'actualise au fur et à mesure ce qui rend l'utilisation de requêtes persistantes une nécessité.

L'évaluation d'une requête continue sur les flux de données de capteurs hétérogènes se fait continuellement. Le résultat des requêtes continues est soit stocké avec des mises-à-jour à l'arrivée de nouveaux éléments, soit restitué à l'utilisateur sous forme de flux.

Un exemple de ces requêtes :

Le flux des moyennes sur 2 minutes des températures relevées par le capteur A.

Ainsi, l'émetteur de cette requête recevra toutes les deux minutes la moyenne de la température relevée par le capteur A. Cette requête s'arrêtera au moment où l'utilisateur décide de le faire. Les requêtes continues sont plus adaptées aux systèmes de gestion de flux de données.

Plusieurs systèmes comme STREAM, TelegraphCQ et SStreamWare [22] ont été proposés dans le domaine de gestion de flux de données et l'évaluation de requêtes continues. Dans le paragraphe suivant, nous allons présenter SSTREAMWARE qui est celui développé au Laboratoire Informatique de Grenoble (LIG).

2.4 SSTREAMWARE : un SGFD générique utilisé dans l'évaluation des requêtes continues

SSTREAMWARE est un intergiciel de gestion de flux de données de capteurs hétérogènes qui a été proposé par le Laboratoire d'Informatique de Grenoble dans le cadre du projet RNRT PISE [22].

Ce système adopte une approche orientée service et une organisation hiérarchique qui permet une évaluation répartie des requêtes.

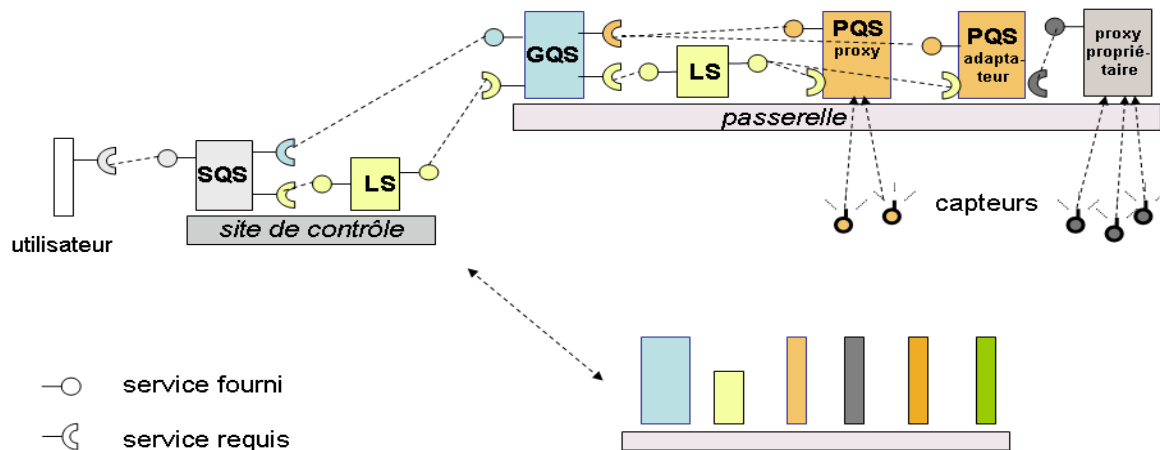


Figure 7 Architecture orientée services dans SSTREAMWARE [22]

La Figure 7 présente les différents niveaux de la hiérarchie :

- les sites de contrôle : auxquels sont adressées les requêtes utilisateurs ;
- les passerelles : ce sont des plateformes logicielles responsables des sous-régions et regroupent plusieurs proxys ou adaptateurs ;
- les proxys ou adaptateurs : ce sont des logiciels généralement propres aux capteurs ;
- les capteurs : responsables de la collecte des données et gérés par leurs proxys.

Les différents services utiles pour l'évaluation des requêtes sont :

- Sensor Query Service (SQS) : ce service est déployé sur les sites de contrôle, il est proposé aux utilisateurs pour formuler leurs requêtes. SQS utilise le Lookup service (LS) pour découvrir les passerelles nécessaires à l'évaluation des requêtes ;
- Gateway Query Service (GQS) : c'est l'équivalent des SQS au niveau des passerelles. Il est chargé d'évaluer les requêtes concernant la sous-région. Le GQS émet ses requêtes aux proxys afin de récupérer les données nécessaires à l'évaluation de requêtes ;
- Proxy Query Service (PQS) : c'est le service utilisé par les proxys pour l'interrogation des capteurs. Son rôle principal est de cacher l'hétérogénéité des capteurs présents dans le réseau.

Dans SSTREAMWARE, les requêtes sont émises au site de contrôle en utilisant soit un mode graphique soit un mode textuel. L'évaluation des requêtes consiste en deux parties : une requête instantanée qui sert à identifier les capteurs concernés par la requête continue et une requête continue qui concerne l'exécution d'opérateurs sur les flux continus. L'utilisation des fenêtres glissantes fait partie des caractéristiques de SSTREAMWARE, toujours dans le but de faire face aux opérateurs bloquants.

Plusieurs propositions ont été faites pour la gestion des données des capteurs. Certaines sont propres aux applications, d'autres sont propres aux capteurs mais les plus intéressantes sont les solutions génériques qui tiennent compte de l'hétérogénéité des capteurs et permettent d'évaluer les différents types de requêtes (instantanée, historique, continue).

Dans la section qui va suivre, nous allons aborder l'aspect qualité de service dans les réseaux de capteurs. Nous allons présenter les différents challenges liés à la QoS ainsi que quelques propositions qui ont été faites pour améliorer cette dernière.

3 Qualité de service dans les réseaux de capteurs

Depuis l'émergence des réseaux de capteurs, les principaux travaux de recherche se sont concentrés sur la gestion des données issues des capteurs, la gestion de l'énergie et le routage, ce qui est important pour la bonne exploitation de ces réseaux [12]. La qualité de service (QoS) n'a pas fait partie des axes d'intérêt des chercheurs dès l'apparition de ces derniers mais le devient de nos jours.

Dans cette partie, il y aura une présentation générale de la notion de la QoS, nous présenterons les différents niveaux de QoS ainsi que les facteurs qui l'impactent. Ensuite, nous détaillerons les différents challenges et les métriques de la QoS dans le contexte des applications à base de capteurs. Enfin, nous présenterons les mécanismes qui peuvent être mis en place afin de garantir un meilleur niveau de service.

3.1 Qu'est ce que la qualité de service

Selon les applications utilisées, les réseaux de capteurs doivent assurer un niveau de qualité de service minimum afin de répondre aux besoins des utilisateurs finaux. Par exemple, dans un système de détection d'incendie, la transmission de l'information doit être faite en temps réel dans le but de mener les actions nécessaires pour contrôler la situation. Le délai de transmission devient moins critique dans le cas d'un système de climatisation qui a la tâche de maintenir une certaine température dans un local.

La notion qualité de service est un ensemble de mécanismes permettant de garantir les bonnes caractéristiques du réseau pour les utilisateurs. Le réseau est composé de plusieurs éléments notamment les équipements actifs et le support de transmission chargé de véhiculer les données. Ces éléments sont appelés ressources du réseau et sont généralement limitées et finies, il faudra donc optimiser l'utilisation de ces ressources pour satisfaire les besoins des utilisateurs et des applications.

La qualité de service a été bien étudiée dans le cadre des réseaux filaires. Dans ce type de réseaux, ce sont généralement les routeurs qui assurent la QoS en utilisant un mécanisme de priorisation des flux de données. Selon la priorité du paquet reçu, le routeur peut soit l'acheminer soit l'ignorer, cela dépendra des ressources disponibles au moment de la réception.

Dans les réseaux de capteurs, le contexte est complètement différent, les ressources de ces derniers sont limitées notamment au niveau de la bande passante, de la capacité de calcul, etc. De plus, si l'on considère que les nœuds peuvent être mobiles, ceci rend plus difficile la transmission des données et par conséquent la qualité de service devient difficile à garantir.

3.2 Niveaux de service et facteurs impactant la QoS

3.2.1 Niveaux de service

Le terme niveau de service dans un réseau de capteurs définit le niveau d'exigence pour la capacité d'un réseau à fournir un service de bout-en-bout. On distingue généralement trois niveaux de QoS [7]:

- Service garanti : aucune dégradation du service fourni n'est permise. En anglais, ce niveau de QoS est connu sous le nom de « Hard QoS ». La QoS avec service garanti est utilisée principalement avec les applications critiques.

- **Service différencié** : à ce niveau, la QoS se base sur la nature des données échangées. Par exemple, si les données concernent la détection d'un événement critique, le niveau de QoS sera du type garanti, si les données concernent une mise à jour ou autre, le niveau sera moins important. Les différents niveaux seront définis au niveau de l'application. Le niveau de service différencié est connu en anglais sous le nom de « Soft QoS ».
- **Meilleur effort** : connu en anglais sous le nom de Best effort ou « Lack of QoS » ne permet d'avoir aucune garantie. Ce niveau de QoS est utilisé dans les environnements non-critiques.

3.2.2 Facteurs impactant la QoS

Ces facteurs sont multiples et variés et dépendent de l'environnement dans lequel sont déployés les capteurs et des capteurs eux-mêmes. Certains de ces facteurs sont d'une haute importance et peuvent même être assimilés à des éléments de mesures de la QoS. Nous en parlerons dans la deuxième partie de ce paragraphe. Nous commençons par la première famille de ces facteurs :

- **Le matériel**

Les ressources limitées des capteurs notamment les unités de calcul peuvent restreindre la capacité du réseau. En effet, dans un environnement physique riche en événements, le capteur doit exécuter un nombre important d'instructions mais dans certains cas ce dernier se trouve dépassé. La non-détection d'un événement peut aussi être causé par un capteur qui peut être soit défaillant, soit à faible capacité de détection. En d'autres termes, il faut utiliser les bons capteurs pour les bonnes applications, sinon on peut être confronté à des erreurs de mesures ou d'observation. Les capteurs doivent aussi s'adapter à l'environnement physique dans lequel ils sont implémentés.

- **L'hétérogénéité des composants dans un même réseau**

Dans un même réseau, on peut rencontrer différents types de capteurs avec différents logiciels et fonctionnalités. Ceci rend la communication entre eux parfois difficile et la qualité de service ne peut plus être garantie comme dans un réseau homogène qui utilise les mêmes protocoles.

- **La mixité du trafic**

Dans certains cas, le trafic sur le réseau peut concerner plusieurs applications et les données sont de types différents. Par exemple, dans une maison équipée d'un système de détection d'incendie et un système de surveillance vidéo, les flux de données et leurs criticités seront totalement différents. La qualité de service dans ce type d'environnement doit prendre en compte ces paramètres et les flux de données doivent être différenciés.

- **La puissance du signal et le support de transmission**

Un capteur qui n'arrive pas à faire parvenir les informations relevées à la station de base ou à un capteur voisin peut être la cause de l'insatisfaction de l'utilisateur. En effet les capteurs doivent être équipés de modules de transmission adaptés à l'environnement, autrement le signal sera perdu en cours de route et l'information n'arrivera pas à destination.

- **La sécurité et le stockage des données**

Les réseaux de capteurs doivent utiliser des mécanismes de sécurité afin de protéger les données transmises. La façon avec laquelle sont gérées les données peut impacter la QoS car ceci peut impacter la quantité d'énergie consommée, la charge du réseau, etc.

Les premiers facteurs que nous venons de présenter ont certes leurs influences sur les applications mais il existe d'autres facteurs qui peuvent être considérés comme des métriques de la qualité de service. Nous les listons ci-dessous.

- **La topologie dynamique des réseaux**

Dans certains domaines, les applications exigent une certaine mobilité des capteurs par exemple dans le transport ou la surveillance des animaux. La topologie peut être amenée à changer en ajoutant ou supprimant des capteurs. La qualité de service doit prendre en compte ces paramètres, le réseau doit continuer à fournir le service demandé dans les meilleures conditions.

- **La couverture réseau**

La couverture réseau est un facteur important dans la qualité de service d'un réseau de capteurs. Par exemple, dans le cas des applications orientées détection d'événements ou de cibles, la non-détection ou la mauvaise interprétation des informations issues de l'environnement physique peut être causée par plusieurs raisons en particulier la topologie du réseau et le déploiement des capteurs. Dans une zone non couverte de capteurs actifs, on peut passer à côté d'un événement majeur. Le nombre de capteurs actifs peut donc être considéré comme un facteur important dans son impact sur la QoS [13].

- **La tolérance aux pannes**

Dans certaines situations, les capteurs peuvent tomber en panne et leur remplacement s'avère impossible. Le réseau doit continuer à fonctionner sans dégradation de service.

- **L'énergie**

L'énergie des capteurs est dissipée pendant la phase de collecte et de transmission des informations. Cette ressource précieuse est limitée et quand un nœud consomme la totalité de sa batterie, il devient hors service et la zone couverte par ce dernier ne l'est plus [13].

3.3 Les métriques dans la qualité de service

Pour répondre aux différents challenges, plusieurs métriques sont prises en compte pour mesurer la QoS. Les principales métriques de la qualité de services sont la bande passante, le délai de transmission, la gigue, le taux de perte de paquets [12] et la durée de vie du réseau.

- **La bande passante**

La bande passante dans un réseau correspond au nombre de bits qu'on peut émettre pendant un intervalle de temps, elle est généralement exprimée en bit par seconde (bit/s). Plus la bande passante est grande plus les performances du réseau sont meilleures.

Les capteurs générant un flux de données important ont besoin d'une large bande passante pour pouvoir transmettre leurs données. Comme exemple, un système de surveillance vidéo a besoin d'envoyer des paquets de données vidéo et ceux-ci nécessitent une bande passante assez large.

- **Le délai de transmission**

Le délai de transmission ou latence est le temps écoulé par un paquet entre son départ du capteur source jusqu'à son arrivée au capteur destination. Ce temps inclut le temps de commutation si le paquet doit traverser plusieurs stations, le temps d'attente si le paquet doit faire la queue à l'arrivée et le temps de propagation qui dépend de la bande passante.

Dans le cas d'une application critique, la transmission des données doit être faite en temps réel, par conséquent le délai de transmission doit être quasi nul.

- **La gigue**

Ce terme signifie la variation du délai de transmission au fil du temps entre des paquets émis par un même nœud. Cette variation peut être due au temps d'attente, au temps de propagation, etc.

Les systèmes nécessitant un niveau de QoS important n'acceptent pas les vibrations de transmission, les données doivent arriver au même rythme.

- **Le taux de perte des paquets**

Le taux de perte des paquets représente le pourcentage des paquets perdus pendant le processus de transmission. Plusieurs facteurs peuvent causer cette perte en particulier une mauvaise connectivité entre les nœuds. Ce critère peut être utilisé pour mesurer la fiabilité du réseau.

3.4 Les mécanismes pour assurer la qualité de service

A cause de la complexité des réseaux de capteurs, les mécanismes mis en place afin de garantir la qualité de service sont divers. Parmi ceux là, on trouve ceux qui s'intéressent à la gestion des ressources du réseau et ceux qui s'occupent de la gestion des données.

3.4.1 Mécanismes orientés réseau

Les protocoles de routage déjà utilisés dans le cadre des réseaux filaires et ad hoc ne peuvent pas être utilisés dans les réseaux de capteurs à cause de la nature et les spécificités de ces derniers. Il a fallu développer d'autres protocoles qui supportent la QoS et qui prennent en considération toutes leurs particularités.

Ces protocoles doivent garantir la communication en temps-réel, l'intégrité des données et la priorisation des flux.

Parmi ces protocoles, nous pouvons citer :

- Le protocole SPEED qui est basé sur l'optimisation des temps de communication de bout en bout. Celui-ci est utilisé pour les applications qui nécessitent une communication en temps réel.

- Le protocole MMSPEED (multi-path multi-speed) qui est une évolution de SPEED. Il offre plusieurs vitesses de transmission et peut établir plusieurs chemins vers une destination. Ce protocole peut être utilisé dans des environnements hétérogènes où le trafic est mixte. Chaque vitesse offerte permet de définir un niveau de QoS et la multitude des routes permet une meilleure qualité du trafic.

3.4.2 Mécanismes orientés données

Plusieurs mécanismes de qualité de service orientés données ont vu le jour afin de palier aux différents problèmes que peuvent rencontrer les réseaux de capteurs. Les problèmes les plus connus sont la précision des mesures, la complétude des données et les données manquantes.

Parmi ces mécanismes, nous pouvons citer :

- Précision des mesures : les capteurs doivent être adaptés aux applications utilisées et à l'environnement dans lequel ils sont déployés.
- L'agrégation des données : ce mécanisme a pour but de limiter le nombre de paquets échangés sur le réseau et par conséquent le risque de perdre les données est minimisé.
- La compression des données : ce mécanisme utilise une méthode de codage des données à la source (capteur) et de leur décodage au niveau de la station de base. Ainsi, la taille des paquets est réduite et leur transmission utilise moins de bande passante.
- L'approche cluster : cette approche permet à la fois de conserver l'énergie des capteurs et limiter les distances parcourues par les paquets. Ceci permet la limitation des pertes des paquets et de garantir un temps de transfert constant.
- Traitement local des données : cette méthode permet de conserver l'énergie des capteurs puisque le module de traitement consomme beaucoup moins d'énergie que le module de transmission ce qui implique une durée de vie du réseau plus longue.

La qualité de service dans les réseaux de capteurs doit prendre en compte plusieurs éléments en commençant par le type d'application et sa criticité, le milieu physique dans lequel sont déployés les nœuds et enfin leur hétérogénéité. Ces éléments réunis ont conduit à définir plusieurs niveaux de service. La QoS peut être impactée par plusieurs facteurs à différents niveaux. Ces facteurs sont présents soit au niveau du capteur lui-même, soit au niveau du réseau mais aussi au niveau de l'application.

Jusqu'à présent les mécanismes qui sont mis en place pour assurer la QoS sont plutôt orientés réseau et données. Ceci reste insuffisant, c'est pour cela que les recherches continuent et d'autres mécanismes pourront voir le jour.

Les capteurs sont connus pour leurs ressources limitées surtout au niveau énergétique. Pour un réseau de capteurs, l'énergie représente sa durée de vie. Ce point est aujourd'hui au cœur des recherches afin de trouver des solutions à la bonne gestion de cette ressource. Nous allons présenter les différents éléments qui participent à la dissipation de l'énergie ainsi que quelques techniques qui ont été proposées pour la conserver dans la dernière partie de cet état de l'art.

4 Consommation et conservation d'énergie

L'énergie est la ressource la plus critique dans les réseaux de capteurs. En effet, cette dernière représente la durée de vie du réseau puisqu'un capteur qui épuise son énergie n'est plus en mesure d'assurer les tâches dont il est responsable.

Cette section est divisée en deux parties, la première présente l'aspect consommation d'énergie ainsi que les différents facteurs qui y participent et la deuxième présente les différentes méthodes que nous pouvons utiliser afin de conserver cette ressource cruciale.

4.1 Consommation d'énergie dans les réseaux de capteurs

Les capteurs sont généralement équipés de batteries qui représentent leur seule source d'énergie. La dissipation d'énergie dans les réseaux de capteurs se fait à travers plusieurs opérations. Il s'agit principalement des opérations de capture (mesure), de traitement et de transmission de données. D'autres facteurs (comme l'écoute du canal, l'échange des paquets de contrôle, etc.) participent à la consommation de l'énergie mais ils sont de moindre importance.

4.1.1 Principaux facteurs dans la consommation d'énergie

4.1.1.1 *Energie consommée par le module de mesure*

L'énergie dissipée par cette opération représente un faible pourcentage de l'énergie totale consommée par un nœud capteur. Les tâches réalisées par l'unité de mesure qui provoquent la consommation de l'énergie sont principalement les suivantes : l'échantillonnage, la conversion des valeurs physiques en valeurs électriques, le traitement des signaux et la conversion des signaux analogiques en signaux numériques.

4.1.1.2 *Energie consommée par le module de traitement*

Au niveau du module de traitement des données, l'énergie est consommée suite à l'exécution d'instructions pour effectuer des calculs. Généralement, le traitement des informations se fait en local pour minimiser les échanges radios entre les nœuds.

Quand ce module n'effectue aucun traitement, une petite quantité d'énergie est consommée, c'est ce qu'on appelle l'énergie de fuite.

4.1.1.3 *Energie consommée par le module de transmission*

Le module de transmission de données (radio) peut avoir l'un des statuts suivants :

- Statut émission : la radio est en train d'émettre des données ;
- Statut réception : la radio est en train de recevoir des données ;
- Statut écoute ou actif (mode idle): la radio n'est ni en train d'émettre ni de recevoir des données mais en train d'écouter le canal de transmission pour détecter d'éventuels paquets circulants sur le réseau.
- Statut endormi : la radio est au repos, elle est mise hors tension.

Les modes émission et réception sont les deux modes où l'on consomme le plus d'énergie. L'énergie consommée lors de ces deux opérations est déterminée par la quantité de données à transmettre et la distance entre les nœuds participants. En effet, plus la puissance du signal est élevée plus la quantité d'énergie consommée sera importante.

Une autre forme de dissipation d'énergie au niveau du module radio apparaît quand ce dernier passe d'un statut à un autre, ceci est dû à l'activité des circuits électroniques. Quand la radio passe par exemple du mode écoute au mode émission pour envoyer des paquets de données, une certaine quantité d'énergie est consommée pour démarrer l'émetteur.

Des études ont montré que la quantité d'énergie consommée en mode *idle* est relativement importante [14], elle est même comparable à celle consommée en mode réception. Il sera donc plus judicieux de ne pas utiliser ce mode et passer directement en mode sommeil. Cet aspect n'est pas encore pris en compte par certains constructeurs.

Le module de transmission représente la source de consommation d'énergie la plus importante dans un nœud capteur puisqu'il a la responsabilité d'assurer la communication entre les nœuds. En effet des expérimentations ont montré que l'énergie utilisée pour transmettre un bit est équivalente à celle utilisée pour exécuter un millier d'instructions au niveau du module de traitement.

4.1.2 Facteurs induisant une surconsommation d'énergie

Bien que la consommation d'énergie lors des opérations de capture, de traitement et de transmission de données soit justifiée, il existe d'autres facteurs qui participent à la dissipation de l'énergie mais qui sont de moindre utilité. La surconsommation d'énergie inclut toute consommation inutile que l'on peut éviter dans le but de conserver l'énergie des capteurs [15]. Ces facteurs sont :

- **La retransmission suite aux collisions**

Les capteurs partagent généralement le même canal de transmission pour l'échange des données. Quand plusieurs capteurs se mettent à transmettre leurs trames en même temps, il peut avoir des collisions entre celles-ci. En présence de ce phénomène, les trames en question sont perdues et doivent être retransmises. La retransmission des données aura un coût supplémentaire significatif en termes d'énergie.

- **L'écoute du canal**

Un capteur en mode écoute active du canal dans l'attente d'un paquet qui lui est destiné et qui ne reçoit rien consomme une certaine quantité d'énergie. Une alternative qui peut remédier à ce genre de problème est de passer la radio en mode sommeil du moment où on sait qu'il y aura pas une grosse activité réseau. Cette solution n'est pas complètement efficace puisqu'on sait que quand le capteur aura besoin de passer en mode écoute ou en mode transmission, il consommera alors une quantité d'énergie significative. La fréquence de changement d'un mode à l'autre doit alors rester raisonnable.

- **La sur-écoute**

Ce phénomène connu sous le nom de « *overhearing* » est expliqué par le fait qu'un capteur reçoit des paquets qui ne lui sont pas destinés. L'*overhearing* conduit à une perte d'énergie conséquente surtout quand on est dans le cas d'un réseau contenant un nombre important de capteurs et où le trafic est important.

- **La surémission**

Quand un capteur envoie des données à un autre et ce dernier n'est pas prêt à les recevoir, les paquets envoyés seront perdus. Dans ce cas, le capteur renverra les données et par conséquent consommera de nouveau une quantité d'énergie. Ce phénomène connu sous le nom d'*overmitting* est fréquent dans les réseaux de capteurs.

- **Les paquets de contrôle**

Les paquets de contrôle sont utilisés pour accomplir les tâches suivantes : signalisation, connectivité et l'évitement des collisions dans le réseau. L'envoi, la réception et l'écoute de ces paquets nécessitent des ressources additionnelles en énergie. Ce phénomène est connu sous le nom d'*overheating*.

- **La distance entre les capteurs**

Un capteur qui a besoin d'échanger des données avec un autre sera obligé de gérer la puissance du signal en fonction de la distance qui le sépare du deuxième capteur. Deux cas se présentent :

- Si la puissance du signal est suffisante, les paquets arriveront à leur destination. La quantité d'énergie consommée peut être importante mais il n'y a pas de perte de données.

- Si la puissance du signal est insuffisante, les paquets n'arriveront pas à leur destination et les données seront perdues. Dans ce cas la perte concerne l'énergie consommée et les données perdues.

- **Autres sources de surconsommation d'énergie**

Il existe d'autres sources de consommation d'énergie dans les réseaux de capteurs telles que :

- Le premier démarrage du capteur : en effet ce dernier va contrôler toutes ces unités.
- L'autodécharge : indépendamment du mode de fonctionnement du capteur (transmission, capture, écoute et sommeil), les batteries se déchargent d'elles mêmes et perdent de leurs capacités au fil du temps.

L'énergie dans les réseaux de capteurs est dissipée de plusieurs façons, il fallait donc trouver des solutions pour la conserver. Dans la partie suivante, nous allons présenter quelques mécanismes qui sont mis en place afin d'atteindre ce but.

4.2 Conservation d'énergie dans les réseaux de capteurs

Comme nous l'avons remarqué, ce sont les tâches liées à la transmission de données qui sont les plus consommatrices de l'énergie comparativement aux autres tâches. Nous allons présenter quelques techniques qui ont été développées pour minimiser la consommation d'énergie soit au niveau capture, soit au niveau traitement soit au niveau transmission. Nous allons voir les mécanismes de base pour la conservation d'énergie, les techniques orientées réseau et en fin nous présenterons les techniques orientées données.

4.2.1 Mécanismes de base pour la conservation d'énergie

4.2.1.1 Mode économie d'énergie

Ce mode, compatible avec n'importe quelle couche MAC (Media Access Control) consiste à passer en mode OFF le module de communication pendant les périodes où le capteur n'envoie ni reçoit des messages [14]. Cette solution est efficace dans le cas où on est en présence de réseau à faible trafic. En effet, il faut faire la comparaison entre le gain en énergie obtenu en éteignant la radio et la quantité consommée lors de son démarrage.

4.2.1.2 Limitation des accusés de réception

En présence d'un réseau à forte densité, il est préférable de limiter les échanges de données afin d'éviter les collisions et les interférences. La limitation des échanges ne va pas concerner les paquets de données mais plutôt les accusés de réception.

4.2.1.3 L'approche cluster

L'approche cluster dans les réseaux de capteurs permet une meilleure répartition de consommation d'énergie entre les nœuds et une meilleure gestion des données [6, 16].

Le principe du clustering consiste à diviser la zone d'activité des capteurs en plusieurs zones, chaque zone est occupée par plusieurs capteurs dont un qui assure le rôle de clusterhead (CH). C'est ce qu'on appelle cluster ou groupement.

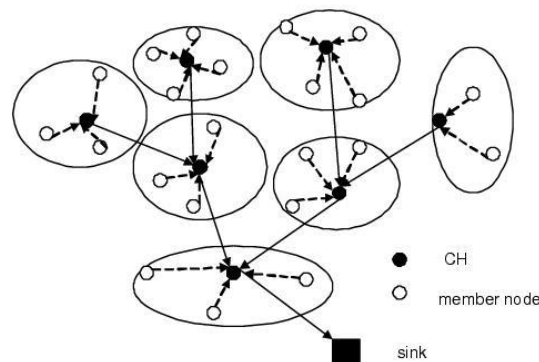


Figure 8 Architecture basée sur les clusters

La Figure 8 représente un réseau de capteurs avec une architecture basée sur les clusters. Les nœuds ne communiquent pas directement entre eux, ils doivent passer par le CH, cette opération est appelée communication inter-cluster [6]. De même, les CH ont la charge d'assurer l'échange des données entre deux capteurs appartenant à deux clusters différents, c'est ce qu'on appelle communication intra-cluster [6]. L'échange avec la station de base entre dans le cadre de la communication intra-cluster.

Afin de minimiser la consommation d'énergie, on change régulièrement le CH. Ainsi, on obtient un réseau dans lequel tous les capteurs auront à peu près le même niveau de batterie.

Cette technique permet aussi de cloisonner le réseau, ainsi on limite les pertes d'énergie liées aux collisions, interférences et retransmission de données.

4.2.2 Techniques orientées réseau

4.2.2.1 Protocoles sommeil/ Réveil

Ce type de protocole appelé aussi *duty cycle* concerne généralement le module radio du capteur [15]. Le principe de base utilisé est qu'un nœud ne se réveille que lorsqu'il doit envoyer ou recevoir des données, il alterne alors entre période de sommeil (sleep) et période de réveil (wakeup). Les protocoles Sleep/Wakeup se divisent en trois familles : à la demande, rendez-vous programmé ou régime asynchrone [17].

- Les protocoles à la demande : le problème qui se pose dans ce type de fonctionnement est de savoir comment informer un capteur qu'il doit se réveiller parce qu'il y a un autre qui veut communiquer avec lui. Parmi les solutions proposées, l'utilisation de capteurs équipés de deux modules radio. Un premier à faible consommation d'énergie et faible débit qui sert à la signalisation et un deuxième pour l'échange des données. On peut citer comme exemple le protocole STEM (Sparse Topology and Energy Management) ;
- Les protocoles basés sur les rendez-vous programmés : un plan de rendez-vous est établi à l'avance. Les capteurs se réveillent à un instant t et restent actifs durant la durée de l'échange des données puis se rendorment jusqu'au prochain rendez-vous. Un exemple de ces protocoles est le protocole ASSLEEP ;
- Les protocoles asynchrones : les capteurs se réveillent quand ils ont besoin d'échanger avec leurs voisins et restent réveillés jusqu'à ce qu'ils établissent la communication. Ce type de protocole est facile à implémenter mais présente des problèmes de latence.

4.2.2.2 Protocoles du niveau MAC

La plupart de ces protocoles se basent sur l'attribution d'un temps d'accès au canal aux capteurs pour échanger leurs données. Trois familles se distinguent : celles basées sur TDMA (Time Division Multiple Access), celles utilisant la contention et les hybrides.

- Protocoles MAC basés sur TDMA : ces protocoles utilisent le mode de fonctionnement TDMA, c'est-à-dire, le temps est divisé en trame et la trame est divisée en petits intervalles.
A chaque nœud est attribué un certain nombre d'intervalles pendant lesquels il échange les données, le reste du temps le capteur passe en mode sommeil. Un exemple de ces protocoles est Energy-aware TDMA-based MAC [18].
Les protocoles MAC basés sur TDMA sont efficaces en conservation d'énergie mais présentent plusieurs inconvénients. En effet, ces derniers utilisent des algorithmes complexes pour la répartition du temps, requièrent une synchronisation fine et sensible aux interférences.
- Protocoles MAC avec contention : le principe est de combiner le protocole MAC de base à un autre protocole Sleep/Wakeup. Les protocoles les plus connus utilisant cette technique sont B-MAC pour MAC Berkeley, S-MAC pour Sensor MAC et T-MAC pour Timeout MAC. Ces derniers sont plus consommateurs en énergie par rapport à ceux basés sur TDMA à cause de la contention mais peuvent être adaptés lors du passage à l'échelle.

- Protocoles MAC hybrides : ces protocoles tentent de combiner les points forts des deux types de protocoles cités précédemment. On peut citer comme exemple le protocole Z-MAC et G-MAC.

4.2.3 Techniques orientées données

Les techniques de conservation d'énergie orientées réseau ne tiennent pas compte de la nature et de la quantité de données détectées et traitées par les capteurs. Par conséquent des techniques orientées données ont été conçues pour minimiser la consommation d'énergie dans les réseaux de capteurs. Les techniques orientées données sont les suivantes :

- **Limitation des redondances des données**

Souvent les données détectées par les nœuds sont redondantes dans le temps. Malgré la faible consommation d'énergie du module de détection, le phénomène de redondance affecte d'une manière indirecte cette précieuse ressource. En effet, les données collectées vont être traitées et transmises à travers le réseau ce qui implique un coût supplémentaire et inutile en énergie.

- **Agrégation des données**

L'agrégation des données a pour but de réduire le volume de données circulant sur le réseau. Le principe utilisé est qu'à chaque fois qu'un nœud reçoit des paquets en provenance d'un nœud voisin, ce dernier effectue un calcul comme par exemple calculer la moyenne entre les données reçues et les siennes, ensuite c'est le résultat de cette opération qui est transmis. Ainsi, la station de base ne recevra qu'un résumé des informations collectées par les capteurs [15]. Les applications ne permettant pas une imprécision au niveau des mesures ne peuvent pas adapter ce genre de mécanisme.

- **Compression des données**

La compression des données consiste à encoder les données au niveau des capteurs afin de réduire la taille des messages et de les décoder au niveau du puits. Parmi les techniques utilisées, nous pouvons citer Lempel-Ziv-Welch (LZW) [23].

Dans les réseaux de capteurs, l'énergie représente la durée de vie du réseau. Cette ressource est dissipée sous plusieurs formes et à différents niveaux de l'architecture.

Le module de transmission se révèle le plus gourmand en énergie suivi du module de détection puis celui de traitement.

Les différentes techniques orientées réseau et données présentées dans les paragraphes précédents offrent des solutions à cette problématique mais ceci reste insuffisant ce qui explique la poursuite des recherches dans ce domaine.

5 Conclusion

Grace au progrès spectaculaire de la technologie, les capteurs n'ont cessé de se développer ces dernières années. Les applications basées sur les réseaux de capteurs sont nombreuses et se multiplient jour après jour. Elles sont présentes dans de nombreux domaines, par exemple, la mesure de la température ou de la pression dans un endroit, la surveillance médicale, les applications militaires, etc.

Les données issues des capteurs sont de natures très hétérogènes et leur gestion semble compliquée. Plusieurs systèmes de gestion de données ont vu le jour suite aux nombreuses recherches qui ont été menées dans ce domaine. L'aspect qualité de service, tolérance aux pannes et énergie reste un point crucial dans les réseaux de capteurs. Nous avons présenté dans cet état de l'art les réseaux de capteurs, les solutions courantes dans la gestion des données, la qualité de service ainsi que l'aspect énergie.

Dans le chapitre suivant, nous allons présenter le matériel sur lequel nous avons travaillé durant le stage, les protocoles et mécanismes utilisés.

Chapitre 2 Le kit Sensinode : Technologie, Matériel et Application

Dans ce chapitre, nous allons présenter dans la première section l'entreprise Sensinode ainsi que leur technologie proposée dans le monde de l'internet des objets. Dans la section 2, nous détaillerons le protocole de la couche MAC, le protocole réseau et le protocole de routage utilisés. Nous présenterons dans la section 3 les Nanosensors qui sont les capteurs commercialisés par Sensinode, leurs compositions et leurs fonctionnalités. Enfin, dans la section 4, le fonctionnement et les interfaces de l'application Nodeview qui sert à gérer les données issues des capteurs N741 de Sensinode.

1 Sensinode

Sensinode est une entreprise d'origine finlandaise, basée aussi en Californie qui produit des solutions matérielles et logicielles dans le monde de l'internet des objets [19].

La vision de Sensinode est que l'internet des objets est la prochaine évolution d'internet. L'internet des objets a pour but d'étendre internet à tout dispositif physique doté d'un système d'exploitation embarqué. C'est la combinaison de l'IP et des services web. Cela permettra la détection en temps réel dans les milieux physiques, le suivi et l'identification dans les entreprises et l'utilisation d'internet dans les réseaux intelligents notamment la maison intelligente (cf. figure 9).

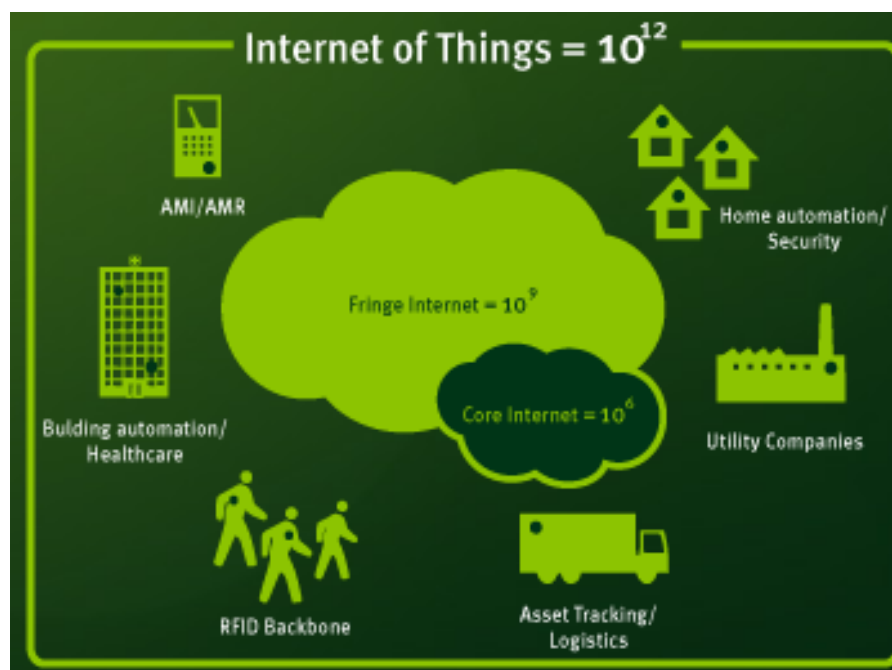


Figure 9 Internet des Objets [19]

Sensinode a conclu des partenariats et collaborations avec des industriels et groupes de standardisation dans ce domaine. Ces principaux partenaires sont :

- Texas Instrument : un des leaders des fabricants de composants électroniques et microcontrôleurs ;
- Radiocrafts : fournit des solutions d'émetteur/récepteur radio destinées à être utilisées dans les réseaux sans fils à faible portée notamment les réseaux de capteurs sans fils ;
- Zigbee Alliance: fournit une solution de communication à faible consommation dans les réseaux de capteurs sans fils ;
- IPSO Alliance (IP for Smart Objects) : cette alliance a été fondée au départ par vingt-sept entreprises opérantes dans le monde de l'internet des objets. Elle est née suite à la demande croissante d'informations et de normalisation dans ce domaine ;
- IETF (Internet Engineering Task Force) : c'est un groupe de travail de dimension internationale qui participe au développement de standards pour le monde de l'internet.

Les différentes solutions présentées par Sensinode sont basées sur la technologie 6LoWPAN « IPv6 over Low Power Personal Area Network » ou « IPv6 for Embedded Device ». Le but est de réduire les tailles des paquets et en-têtes des trames afin de les adapter aux réseaux sans fils à faibles ressources et fortes contraintes (cf. figure 10). Dans la section qui va suivre nous allons présenter cette technologie.

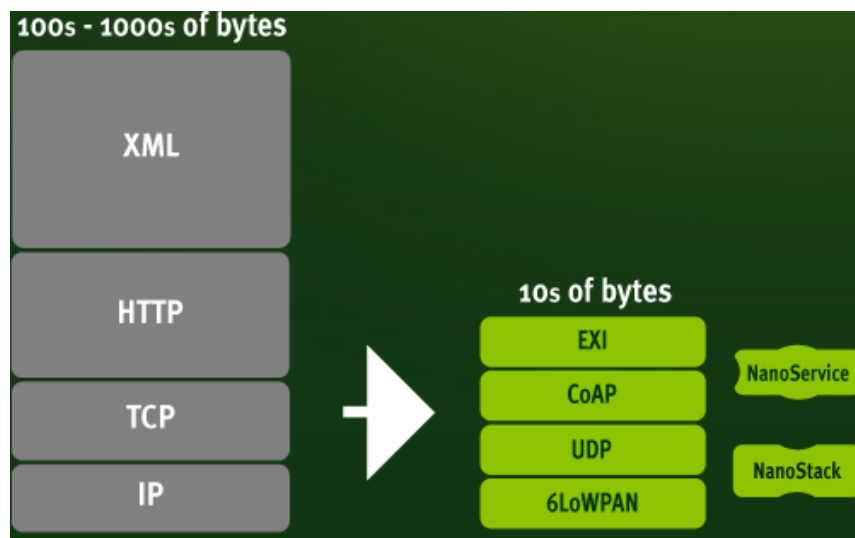


Figure 10 Technologie Sensinode basée sur 6LoWPAN [19]

2 La technologie 6LoWPAN

6LoWPAN est le nom d'un groupe de travail de l'IETF. C'est aussi l'acronyme de *IPv6 Low Power Wireless Personal Area Network* (réseaux sans fils à faible puissance). L'objectif du groupe de travail 6LoWPAN était de définir les mécanismes d'encapsulation et de compression d'en-têtes des paquets IPv6 afin de leur permettre d'être envoyés ou reçus via le protocole de communication IEEE 802.15.4 qui est le protocole utilisé dans les réseaux à faible puissance comme les réseaux de capteurs sans fils. La RFC 4919 et la RFC 4944 présentent la spécification de base développée par 6LoWPAN.

Dans cette section nous allons présenter succinctement les réseaux LoWPAN, les protocoles IPv6 et IEEE 802.15.4 et enfin les mécanismes de base de la technologie 6LoWPAN.

2.1 LoWPAN

Un LoWPAN ou LR WPAN (*Low Rate Wireless Personal Area Network*) est constitué de dispositifs dont les caractéristiques sont une faible consommation d'énergie, une faible portée, un faible débit et un faible coût de production. Les ressources des dispositifs formant ce type de réseaux sont généralement limitées notamment en énergie, mémoire et puissance. L'exemple type de ces dispositifs sont les capteurs. Les nœuds composants un LoWPAN ont pour rôle de collecter et de transmettre des informations vers un équipement central du réseau qui aura la charge de traiter ces informations et les router vers leurs destinations. Cet équipement est appelé puits ou PAN coordinateur.

Dans les réseaux sans fils à faible portée, on distingue deux types de nœuds, les FFD pour *Full Function Device* et RFD pour *Reduced Function Device*. Ces nœuds peuvent avoir les fonctions suivantes :

- Puits : c'est le nœud central du réseau. Il a pour rôle d'initier le réseau, rassembler les informations et les données qu'il acheminera par la suite vers l'extérieur. Les puits sont aussi appelés PAN coordinateur ou gateway et sont de type FFD ;
- Les coordinateurs : ce sont des nœuds de type FFD qui peuvent jouer le rôle de gateway intermédiaire dans le cas où le réseau est composé de plusieurs zones ou clusters. Ils auront la charge d'acheminer les paquets vers la station puits ou vers les autres nœuds enfants ;
- Les nœuds terminaux : ce sont les derniers éléments de la chaîne. Ils ont la tâche de collecter les informations et les transmettre vers leurs parents (nœuds coordinateurs) dans le cas d'un réseau multisauts ou directement vers la station puits (PAN coordinateur) quand il s'agit d'une architecture simple.

Nous pouvons dire qu'un FFD peut jouer le rôle de puits, de coordinateur ou de terminal alors qu'un RFD ne peut assurer que le rôle de nœud terminal ou de capteur dans le cas d'un réseau de capteurs. Les RFD ne peuvent communiquer qu'avec des FFD alors que les FFD peuvent communiquer à la fois avec les RFD et les FFD.

Dans les réseaux LoWPAN, on peut rencontrer trois types de topologie comme le montre la figure 11.

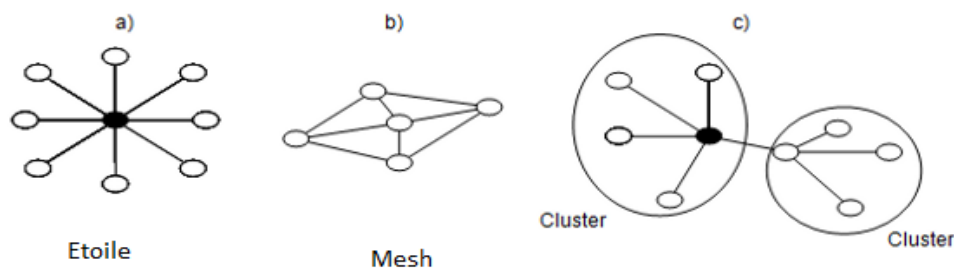


Figure 11 Topologie dans les réseaux LoWPAN

- La topologie en étoile : les échanges se font directement avec le PAN coordinateur. Il n'y a pas de communication possible entre les autres nœuds du réseau. C'est à la station puits d'initier le réseau, d'établir la communication avec tous les dispositifs se trouvant dans sa portée radio;
- La topologie mesh : tous les nœuds peuvent communiquer entre eux. Il s'agit d'un réseau maillé où plusieurs liens sont établis et différents chemins sont possibles ;
- La topologie cluster : à la base c'est une topologie mesh. La différence est que le réseau est hiérarchisé c'est-à-dire qu'il y a un PAN coordinateur, plusieurs coordinateurs responsables chacun d'une zone qui contient un ou plusieurs nœuds terminaux.

Les réseaux LoWPAN utilisent le protocole de communication IEEE 802.15.4 que nous allons présenter dans la section suivante.

2.2 Le standard IEEE 802.15.4

Le standard IEEE 802.15.4 [26] est utilisé au niveau des deux premières couches du modèle OSI, c'est-à-dire la couche physique et la couche MAC.

2.2.1 La couche physique

Au niveau de cette couche, le protocole IEEE 802.15.4 a pour rôle de gérer le support physique sur lequel sont transmis les paquets. Dans la couche physique, le protocole définit les techniques de transformation des bits en signaux numériques et inversement. Les différentes fonctionnalités de cette couche sont :

- La gestion du module radio : on parle de son activation ou sa désactivation mais aussi de son état c'est-à-dire émission, réception ou éteinte ;
- La détection de la puissance du signal (*ED Energy Detection*);
- Fournir des indications sur la qualité du lien radio : cette information est connue sous le nom de LQI (*Link Quality Indication*). Les indications sont fournies à la couche supérieure (MAC) ;
- La détection de trame sur le médium : cette fonctionnalité est assurée par le mécanisme CCA (*Clear Channel Assessment*).

Le protocole IEEE 802.15.4 offre la possibilité d'utiliser deux bandes de fréquences, la bande 868/915 Mhz ou la bande 2450 Mhz (le kit Sensinode utilisé dans ce stage utilise la deuxième bande).

2.2.2 La couche MAC

La couche MAC (Medium Access Control) a pour but la gestion de l'accès au canal. Le mécanisme utilisé pour l'accès au canal est CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Deux méthodes peuvent être utilisées, soit le CSMA/CA Slotted où des périodes de réveil/sommeil sont introduites, soit le CSMA/CA Unslotted où aucun mécanisme de duty cycle n'est déployé. Les fonctionnalités assurées par la couche MAC du protocole IEEE 802.15.4 sont :

- La gestion de l'accès au réseau ;
- La gestion des acquittements ;
- La gestion de l'adressage ;
- La gestion des slots de temps ;
- Le mécanisme de découverte de voisinage.

Ces différentes fonctionnalités sont assurées par des trames spécifiques, on distingue quatre types de trames :

- Les trames Beacon (Beacon frame) qui sont envoyées uniquement par les nœuds coordinateurs et servent à administrer le réseau ;
- Les trames de données (Data frame) servent au transfert des données utiles entre les différents nœuds du réseau ;
- Les trames d'acquiescement (Acknowledgment frame) servent à notifier la bonne réception des trames ;
- Les trames de contrôle (Command frame) servent à effectuer des demandes spécifiques comme par exemple l'association au réseau.

Deux modes de fonctionnement sont possibles avec le standard IEEE 802.15.4. Il s'agit des deux modes balisé et non-balisé.

2.2.2.1 Le mode balisé (*Beacon Enabled Mode*)

Il s'agit d'un mode dit synchronisé parce que tous les nœuds d'un même réseau doivent suivre une structure périodique appelée SuperTrame (*SuperFrame*).

Cette structure périodique commence toujours par l'envoi d'une trame par le coordinateur appelée Beacon. Cette trame Beacon a pour rôle de synchroniser les nœuds avec leur coordinateur, et fournir toutes les informations vitales au fonctionnement du réseau, comme par exemple l'identifiant du réseau, les données en attente au niveau du coordinateur (communication indirecte), le temps où le prochain Beacon est envoyé et la durée de la partie active de la SuperTrame, etc.

Ce mode est celui le plus utilisé car il offre des performances plus importantes en terme de débit, de conservation d'énergie et de fiabilité.

2.2.2.2 Le mode non-balisé (*Non Beacon Enabled Mode*)

Ce mode appelé aussi Unslotted Mode est généralement utilisé en l'absence de contraintes énergétiques, par exemple en cas d'alimentation par secteur.

En utilisant ce mode, il n'y a aucune synchronisation entre les nœuds et pour que ces derniers puissent échanger des données, ils doivent laisser leurs modules de transmission allumés ou se réveiller périodiquement afin d'interroger le nœud coordinateur pour savoir s'il y a des messages qui leur sont destinés.

2.2.3 Le transfert des données

Le standard IEEE 802.15.4 définit trois modes de transfert de données, cela dépend du mode utilisé (balisé ou non-balisé) et de la topologie du réseau.

- **Nœuds terminaux vers coordinateur** : les données sont envoyées de la part des enfants (nœuds terminaux) vers leur coordinateur. Ce mode est utilisé dans la topologie cluster et la topologie en étoile. Si c'est le mode Beacon Enabled qui est utilisé, les nœuds doivent d'abord réceptionner la trame Beacon du coordinateur avant d'envoyer leurs trames en utilisant le CSMA/CA Slotted durant la partie active de la supertrame pour leur transmission. Si au contraire c'est le mode Non Beacon Enabled qui est utilisé, les nœuds envoient directement leurs données en passant par le CSMA/CA Unslotted. Dans ce cas les coordinateurs devront rester réveillés et une grande partie de leurs ressources en énergie est consommée.
- **Coordinateur vers enfants** : Ce mode de transmission est utilisé si on est dans le cas de réseau hiérarchisé (topologie cluster) ou étoile. Si le réseau utilise le mode Beacon Enabled, le coordinateur doit indiquer qu'il y a des trames en attente pour ses fils dans son Beacon, ces derniers savent alors à la réception de ce dernier qu'ils ont une donnée en attente auprès de leur coordinateur et peuvent ainsi en solliciter la transmission. En mode non Beacon Enabled, en l'absence d'envoi périodique des Beacon, les nœuds demandent périodiquement à leur coordinateur s'il y a des données en attente (polling).
- **Transmission de données pair-à-pair** : si le réseau est un réseau de type pair-à-pair (étoile), les nœuds communiquent entre eux en utilisant le CSMA/CA Unslotted. Dans ce cas soit les nœuds restent réveillés tout le temps pour d'éventuelles transmissions, soit ils doivent se synchroniser.

Rappelons que dans la technologie 6LoWPAN, on utilise le protocole IPv6 combiné avec le standard IEEE 802.15.4. Dans la partie qui va suivre, nous allons présenter ce protocole.

2.3 Le protocole Internet (IP)

Avant de parler de IPv6 qui est le protocole internet utilisé dans les réseaux 6LoWPAN, nous allons faire un rappel de IPv4.

2.3.1 IPv4

La version 4 (IPv4) est la première version du protocole internet. Elle a été développée il y a un peu plus de trente ans et a été largement déployée avec l'explosion de la bulle internet au début des années 2000. Elle est décrite dans la RFC numéro 791 (RFC 791).

Internet Protocol représente la couche réseau du modèle OSI dans les réseaux informatiques (cf. figure 12).

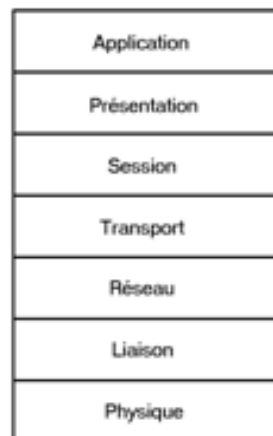


Figure 12 Le modèle OSI

IPv4 utilise un adressage sur 32 bits par conséquent le nombre d'adresses utilisables s'élève à 2^{32} c'est-à-dire 4 294 967 296 adresses.

Une adresse IP dans la version 4 est représentée sous la forme de quatre nombres décimaux séparés par des points par exemple 192.168.0.200. La taille maximale d'une trame appelée MTU (Maximum Transmission Unit) dans les réseaux Ethernet par exemple est de 1500 octets. Pour reconstituer l'information, les équipements se servent des différents champs constituant l'en-tête de la trame IPv4 (cf. Figure 13).

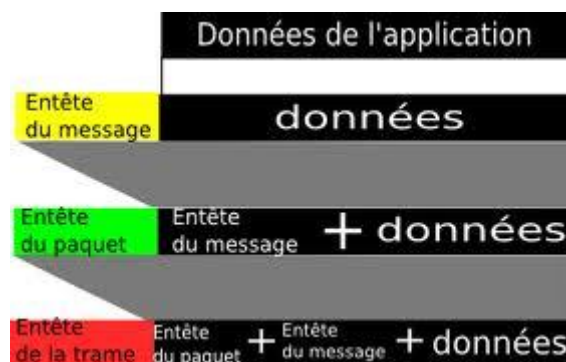


Figure 13 Schéma IPv4

Dans le contexte de l'internet des objets, où chaque équipement doit avoir son adresse IP, IPv4 ne pourra pas répondre aux besoins, c'est pour cela qu'on s'est retourné vers la version 6, IPv6.

2.3.2 IPv6

IPv6 a vu le jour non pas à cause de l'idée de l'internet des objets mais à cause d'un début de pénurie des adresses IPv4.

Les grands principes du protocole IP version 6 sont identiques à ceux du protocole IP version 4. Comme son nom l'indique, IPv6 est une nouvelle version du protocole IP et non pas un nouveau protocole. Malgré tout et fort de l'expérience acquise avec les réseaux IPv4, le protocole IPv6 apporte un certain nombre de nouvelles fonctionnalités.

- Un plus grand espace d'adressage : c'est la plus flagrante évolution mise en avant lorsque l'on parle d'IPv6. Ce plus grand nombre d'adresses disponibles permettra d'adresser et d'identifier tous les équipements communicants. Il permettra de s'affranchir des NAT (Network Address Translation), de déployer de nouvelles applications nécessitant des communications de bout en bout ;
- Un en-tête simplifié et efficace : il est important de constater que l'en-tête du paquet IP a été considérablement simplifié entre les versions 4 et 6 (cf. figure 14). L'en-tête IPv6 est désormais de taille fixe. Les options de l'en-tête IPv4 ont disparu et sont remplacées par des extensions d'en-tête dont le fonctionnement est différent. Dans la version 4, les options étaient analysées par tous les nœuds intermédiaires d'une liaison, les extensions IPv6 ne sont gérées que par les « équipements terminaux ». Le champ « checksum » de l'en-tête IPv4 a lui aussi disparu. Dans une communication IPv4, chaque nœud intermédiaire effectuait une vérification du « checksum » du paquet, décrémenait le champ « Time To Live » et recalculait un nouveau « checksum ». Cette opération se répétait autant de fois qu'il y avait de nœuds, ceci était très pénalisant en termes de ressources. Avec IPv6, on considère que les éléments du réseau introduisent peu d'erreurs dans les communications. La gestion des paquets erronés est déléguée aux équipements d'extrémités et sur les couches supérieures telles TCP et UDP. Les équipements intermédiaires du réseau sont donc déchargés d'une partie des traitements ;

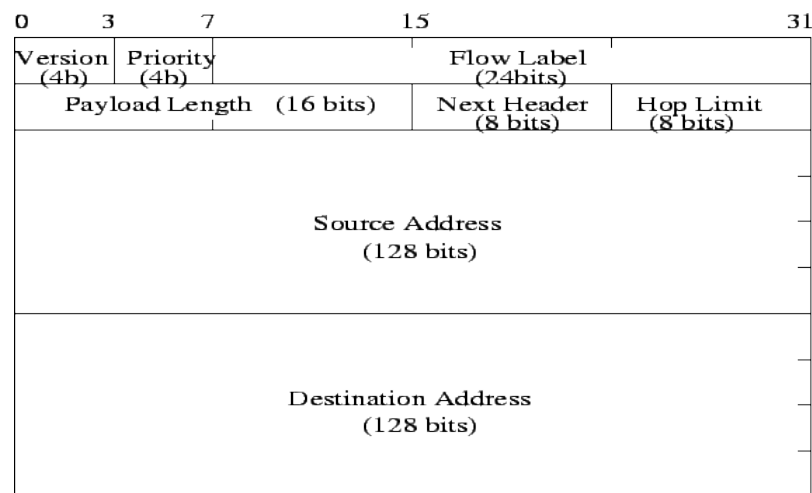


Figure 14 Schéma IPv6

- Autoconfiguration : l'autoconfiguration qui est une des caractéristiques des réseaux de capteurs met en œuvre un certain nombre de nouveaux protocoles associés à IPv6, par exemple le protocole de découverte des voisins, nouvelle version d'ICMP (Internet Control Message Protocol), etc. L'autoconfiguration permet à un équipement de devenir complètement « plug-and-play ». Il suffit de connecter physiquement la machine pour qu'elle acquière automatiquement une adresse IPv6 et une route par défaut. Ceci facilite la renumérotation (ré-adressage des équipements et des machines) ;

- **Mobilité** : la mobilité a été pensée dès la conception du protocole IPv6. Même s'il existe une mobilité IPv4, la mobilité IPv6 est beaucoup plus simple à mettre en place. La mobilité caractérise le fait d'être connecté et de disposer de son environnement tout en se déplaçant et ce, sans interruption de service. La mobilité fait référence à la plupart des objets communicants qui sont ou deviendront des objets « mobiles ». Ces équipements ont besoin d'être connectés quels que soient les lieux où ils se trouvent ou bien durant leur déplacement.

L'adressage dans IPv6 est différent par rapport à IPv4. En effet, ce ne sont plus 32 bits mais 128 soit quatre fois la taille d'une adresse IPv4.

Notation IPv6

L'adresse IPv6 est composée de 128 bits. Cette taille permet de disposer d'un nombre très important d'adresses par rapport à celui offert par IPv4. Avec IPv6, on peut disposer de 10^{38} adresses théoriques. Ceci rend la notation utilisée dans IPv4 inutilisable. On abandonne donc la notation décimale pointée par exemple 192.168.0.200 au profit d'une écriture hexadécimale.

Une adresse IPv6 est composée de huit groupe de deux octets chacun soit 16 bits par mot, les mots étant séparés par un signe deux points « : ».

Exemple d'une adresse IPv6 : 2002:0cb8:0000:43a3:0000:0000:cd11:8001.

Quand un groupe contient des zéros, on peut omettre les zéros non significatifs, l'adresse ci-dessus sera écrite de cette façon 2002:cb8:0:43a3:0:0:cd11:8001.

De la même façon, si on a deux groupes consécutifs de quatre zéros, on peut les omettre entièrement mais on garde les « : » de chaque côté, cette notation est appelé forme abrégée. Ainsi, l'adresse ci-dessus peut être écrite de cette façon 2002:cb8:0:43a3::cd11:8001.

Le protocole IP dans sa version 6 semble répondre à tous les besoins en termes d'adressage dans le monde de l'internet des objets. IPv6 combiné au protocole IEEE 802.15.4 a donné naissance à la technologie 6LoWPAN, nous allons présenter dans le paragraphe qui va suivre les mécanismes de base de cette technologie.

2.4 Fonctionnement de 6LoWPAN

2.4.1 Description

En 802.15.4, la taille maximale du PSDU (Physical layer Service Data Unit) est de 127 octets. Avec les 25 octets de la sous-couche MAC (sans sécurisation), il en reste 102 octets au niveau de la couche liaison. La sécurisation de la couche de liaison de données (AES-CCM-128) nécessite 21 octets, il ne reste que 81 octets disponibles au niveau IP. A ce niveau, il faut déduire 40 octets pour les en-têtes des paquets IPv6. Arrivant à la couche transport, des éventuels en-têtes d'extension, d'UDP (8 octets) ou de TCP (20 octets) sont à prendre en compte.

Finalement les données utiles sont peu élevées 33 octets sur UDP et 21 sur TCP, (cf. figure 15) et ne permettent pas de respecter les spécifications d'IPv6 qui imposent un MTU (Maximum Transmission Unit) minimal de 1280 octets.

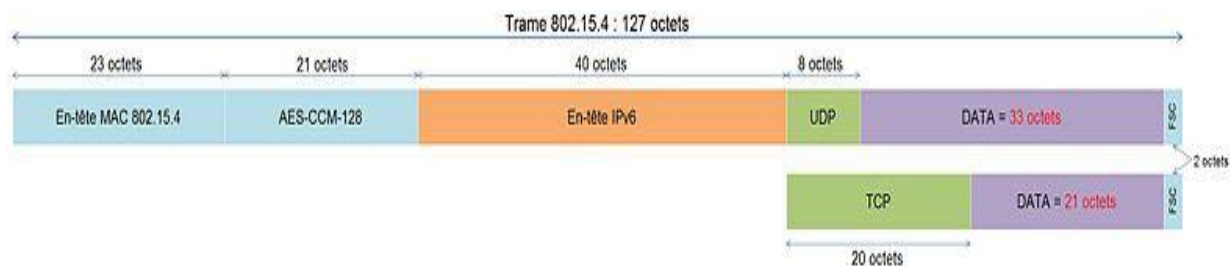


Figure 15 Structure d'une trame 802.15.4

2.4.2 Fragmentation et réassemblage

Afin d'adapter les trames IPv6 au support 802.15.4, des mécanismes de fragmentation au niveau de la source et de réassemblage au niveau de la destination sont utilisés. La source doit fragmenter un paquet IPv6 en plusieurs trames 802.15.4, et l'équipement distant doit réassembler toutes les trames 802.15.4 reçues pour régénérer le paquet IPv6 d'origine. Ces tâches sont consommatrices de ressources (mémoire et CPU) et génèrent de la latence. La latence est due à la file d'attente, le temps de création et de vérification des en-têtes.

Les tâches de fragmentation et réassemblage des paquets sont à la charge de la couche adaptation de 6LoWPAN du modèle OSI (cf. figure 16). Elle reçoit de la couche réseau des paquets IPv6 de 1280 octets et les envoie à son équivalent sur l'équipement distant dans des trames 802.15.4 de 127 octets.

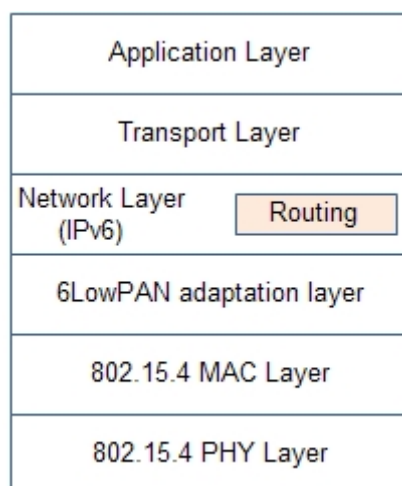


Figure 16 La couche adaptation de 6LoWPAN

Les fragments contiennent des informations de 4 ou 5 octets qui forment l'en-tête de fragmentation. Cet en-tête contient les informations suivantes (cf. figure 17) :

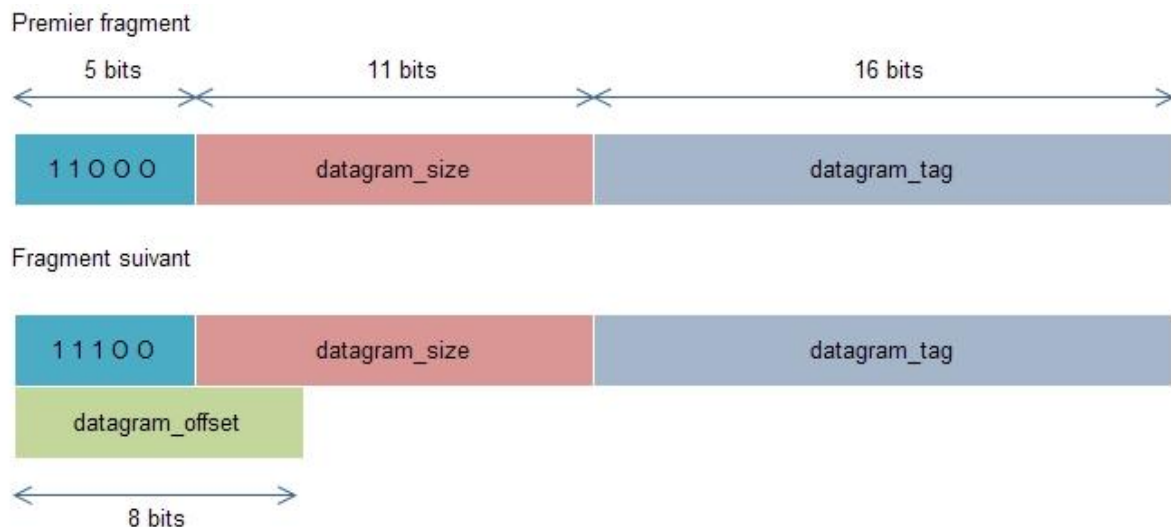


Figure 17 En-tête d'un fragment 6LoWPAN

- Le champ dispatch (5 bits): il s'agit des 5 premiers bits. Ca permet d'identifier qu'il s'agit d'un fragment. Le premier fragment aura la valeur « 11000 » et les suivants « 11100 » ;
- Le champ datagram_size (11 bits): pour la taille du paquet IP avant fragmentation ;
- Le champ datagram_tag (16 bits): c'est un identifiant commun à tous les fragments d'un même paquet IP ;
- Le champ datagram_offset (8 bits) : sert à identifier la position du fragment dans le paquet IP (uniquement présent dans les fragments suivant le premier).

Avec cette technique, si un fragment est perdu, le paquet IP ne peut pas être reconstitué. Le problème avec ce mécanisme de fragmentation est qu'il faudra alors émettre à nouveau tous les fragments. Récemment, des techniques de récupération de fragments perdus ont été proposées [20]. L'idée est d'introduire un nouvel en-tête de fragmentation et surtout un mécanisme d'acquittement des fragments. L'acquittement permet à l'expéditeur de ne retransmettre que les fragments non reçus.

Dans les réseaux LoWPAN, la consommation d'énergie est une contrainte majeure et avec la technique de fragmentation, les échanges sur le réseau se multiplient sans pour autant transporter une quantité considérable de données dans chaque fragment (UDP 33 octets et TCP 21 octets). Il fallait donc trouver des solutions pour limiter les échanges et pouvoir transporter dans chaque paquet transmis un maximum de données. La technique utilisée est la compression d'en-tête.

2.4.3 Compression d'en-tête

Cette technique proposée par le groupe 6LoWPAN s'appelle LoWPAN_IPHC (Lowpan IP Header Compression). Il s'agit de compresser l'en-tête IPv6, l'en-tête du fragment (le dispatch) et l'en-tête UDP. Avec LoWPAN_IPHC, l'en-tête IPv6 peut être réduit à 2 ou à 7 octets :

- Pour les communications sur un lien local l'en-tête IPv6 peut être réduit à 2 octets (1 octet Dispatch et 1 octet LOWPAN_IPHC) ;

- Pour des communications nécessitant plusieurs sauts IP, l'en-tête peut être compressé sur 7 octets (1-octet Dispatch, 1-octet LOWPAN_IPHC, 1-octet nombre de saut, 2 octets pour l'adresse source et 2 octets pour l'adresse destination).

Ainsi, dans une trame 802.15.4 avec la technique de compression d'en-tête, la partie « data » pourra atteindre 75 octets (cf. figure 18).

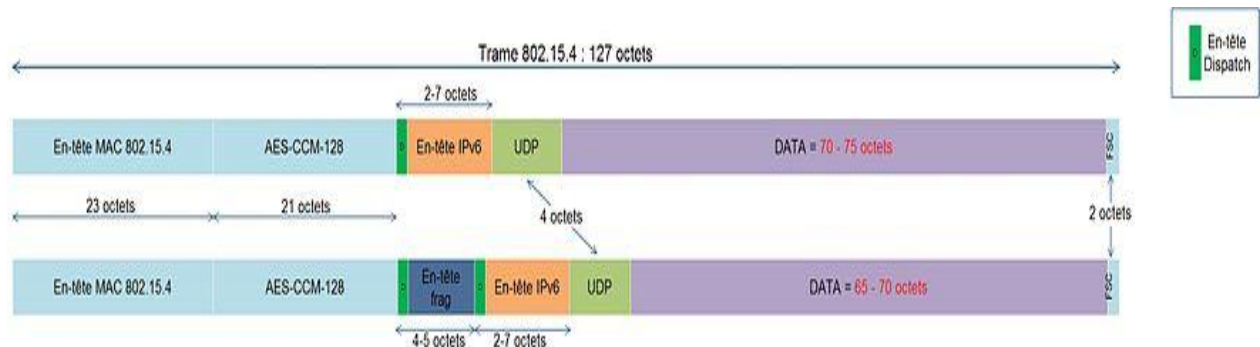


Figure 18 Trame 802.15.4 avec LoWPAN_IPHC

Afin de pouvoir acheminer les trames et paquets dans un réseau LoWPAN, un protocole de routage est nécessaire et doit être adapté à IPv6 et ses fonctionnalités. Il existe deux types de protocoles de routage adaptés à 6LoWPAN :

- Mesh-under : il consiste à implémenter le routage au niveau de la couche adaptation de 6LoWPAN. La décision de routage se fait au niveau 6LoWPAN et donc seulement avec les fragments du paquet IPv6.
- Route-over : le routage est implémenté au niveau de la couche réseau du modèle OSI. Les paquets IPv6 sont reconstitués au niveau de chaque équipement intermédiaire afin de déterminer un nouvel itinéraire de routage. Cette technique permet une détection plus rapide en cas de perte de paquets et une demande de retransmission n'est faite qu'au nœud émetteur.

Les deux techniques sont illustrées dans la figure 19.

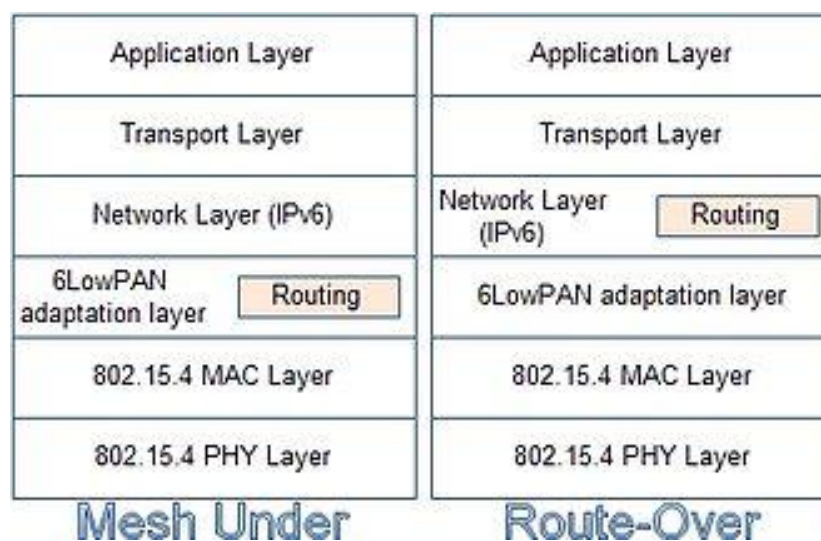


Figure 19 Modes de routage dans 6LoWPAN

La solution adaptée par Sensinode se base sur la technique route-over, il s'agit du protocole RPL.

2.5 Le protocole de routage RPL

RPL est un protocole de routage IPv6 à vecteur de distance, il est implémenté en route-over.

2.5.1 Principe de fonctionnement

Le schéma logique de base consiste à créer un DAG (Directed Acyclic Graph), voir figure 20.

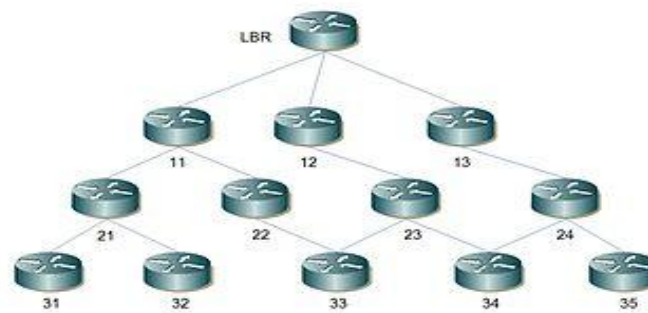


Figure 20 Structure DAG dans RPL

Le LBR (*Low power and lossy network Border Router*) désigne l'équipement de bordure du réseau. Dans un DAG, le LBR est de rang 1. Les équipements de rang supérieur et le LBR forment un DODAG (*Destination Object Directed Acyclic Graph*).

Le LBR commence par envoyer un DIO (*DODAG Information Object*) afin de permettre à chaque nœud dans le réseau de déterminer son rang. Lorsqu'un équipement reçoit une nouvelle version de DIO, il calcule notamment son rang par rapport à celui qu'il vient de recevoir et propage son DIO. Vus d'un nœud, tous les équipements possédant un rang inférieur peuvent prétendre être parents. Les routes optimales au sein d'un DAG sont obtenues à partir de métriques et de contraintes. Le LBR émet périodiquement des DIO pour mettre à jour le DAG. Lorsqu'un équipement rejoint le réseau ou perd le lien vers son « parent », il peut attendre le prochain DIO ou demander l'envoi d'un DIO par un message de sollicitation DIS (*DODAG Information Solicitation*).

2.5.2 Autoconfiguration IP

RPL se base sur les mécanismes ND (*Neighbor Discovery*) et RS (*Router Solicitation*) pour la gestion des adresses IP (cf. figure 21). Les routeurs par défaut gèrent une table NCE (*Neighbor Cache Entry*), où sont listées toutes les adresses du réseau 6LoWPAN. Lors d'une sollicitation, si une adresse n'est pas dans le cache, elle est considérée comme valide et est enregistrée avec une valeur « durée de vie ».

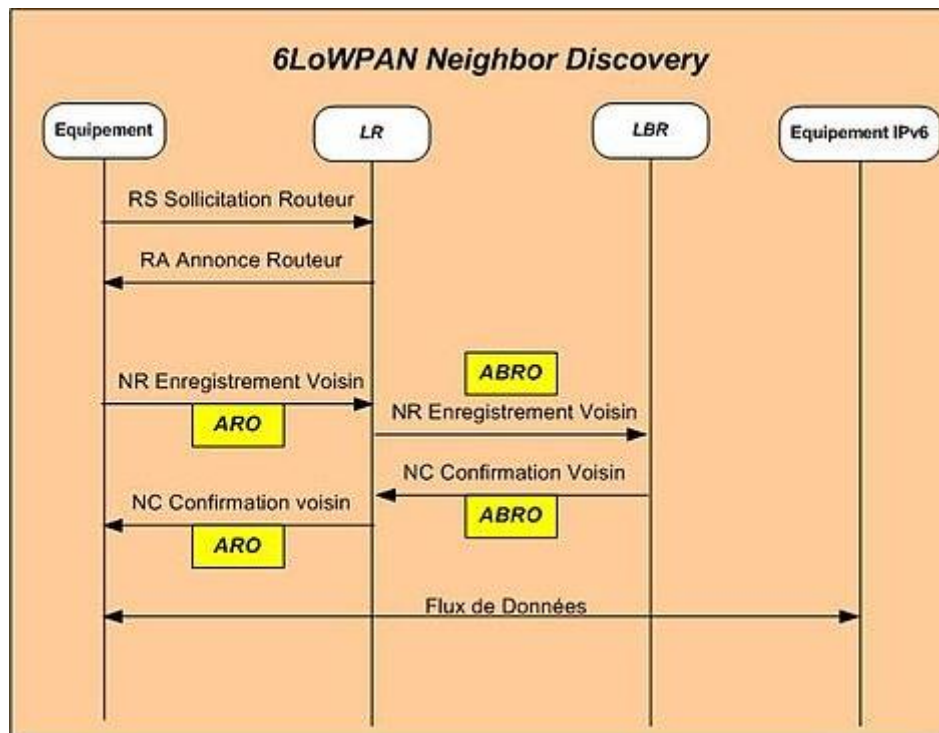


Figure 21 Découverte de voisins dans 6LoWPAN basé sur RPL

Un paramétrage par défaut, adapté aux périodes de sommeil des équipements d'un LoWPAN, permet de conserver leurs adresses valides entre deux réveils. Cette méthode permet de ne pas pénaliser les équipements en consommation d'énergie et évite de relancer une autoconfiguration à chaque réveil. Les nœuds d'un réseau transmettent régulièrement au LBR des ARO (Address Registration Option) contenant la durée de vie de leurs adresses. Le LBR et les routeurs intermédiaires LR (*Low power and lossy network Router*) échangent régulièrement des ABRO (*Authoritative Border Router Option*), ces messages de type RA (*Router Annoucement*) transportent des préfixes d'adresses, des informations de contexte, et l'adresse du LBR.

RPL, 6LoWPAN et 802.15.4 sont les trois protocoles de base des architectures proposées par Sensinode. Le matériel que nous avons utilisé et évalué fera l'objet de la section 3. Dans cette section, nous présenterons les capteurs N741 et l'application Nodeview.

3 Le kit 6LoWPAN de Sensinode

Le kit acquis par l'équipe SIGMA est composé de :

- 1 Nanorouter 2.0 : c'est le routeur IPv6 ;
- 10 Nanosensor plus les antennes : ce sont les capteurs N741 que nous allons présenter en détail dans le paragraphe suivant ;
- 2 câbles USB : ils sont utilisés pour la programmation des capteurs et servent aussi à charger les piles ;
- 20 piles NiMH rechargeables ;
- L'application NodeView 2.0 ;
- Un exemple de code en langage C pour la programmation des capteurs auquel nous apporterons des modifications pour l'adapter à nos besoins.

3.1 Le Nanorouter

Il joue le rôle de LBR dans l'architecture, c'est le PAN coordinateur du réseau. Le Nanorouter peut opérer dans deux bandes de fréquences, celle des 2,4 GHz et celle des 815 MHz. Pour configurer le Nanorouter, il suffit de s'y connecter en Telnet avec le login et le mot de passe fournis par Sensinode. La seule configuration possible est au niveau réseau. Il suffit de configurer son adresse IPv6 pour établir la communication avec le poste de travail sur lequel est implémentée l'application NodeView 2.0. Pour cela, il faut vérifier que le poste de travail et le Nanorouter aient le même *prefix* de l'adresse IPv6, par exemple 2001:470:55:: .

Une fois cette étape effectuée, le Nanorouter apparaîtra dans l'interface *Router View* de l'application NodeView 2.0. L'utilisation de NodeView 2.0 sera abordée dans la prochaine section de ce chapitre.

3.2 Les capteurs Nanosensor N741

Les Nanosensor N741 que nous avons utilisés durant le stage ont les spécifications et fonctionnalités suivantes :

- Ils sont compatibles avec la technologie 6LoWPAN ;
- ce sont des capteurs sans fils équipés de deux batteries NiMH ;
- ils sont équipés d'un capteur de luminosité et d'un capteur d'accélération.

Comme tous les capteurs, les Nanosensor N741 sont composés en plus de leur source d'énergie de trois unités : acquisition, traitement et transmission.

3.2.1 L'unité d'acquisition et de traitement

Il s'agit du microcontrôleur MSP430F5438 de Texas Instrument. Les principales caractéristiques techniques de ce microcontrôleur sont :

- Un microprocesseur à jeu d'instructions réduit (RISC) de 16 bits ;
- Un registre de 16 bits ;
- Deux Timers : Timer A et Timer B ;
- Un convertisseur analogique numérique (ADC) 12 bits ;
- Une mémoire RAM 4 KB ;
- Une mémoire flash de 256 KB ;

- Un fonctionnement dans des températures extrêmes allant de -40 °C jusqu'à 85 °C ;

En plus de ces caractéristiques, le MSP430f5438 peut opérer dans cinq modes différents. Pour chacun des modes, une ou plusieurs fonctionnalités, ou composants sont désactivés afin de moins consommer d'énergie.

Les cinq modes opératoires sont :

- Mode actif (Active mode, AM) : c'est le mode le plus consommateur d'énergie, toutes les horloges sont actives ;
- Mode Consommation réduite 0, LPM0 (Low Power Mode 0) ;
- Mode Consommation réduite 1, LPM1 (Low Power Mode 1) ;
- Mode Consommation réduite 2, LPM2 (Low Power Mode 2) ;
- Mode Consommation réduite 3, LPM3 (Low Power Mode 3) ;
- Mode Consommation réduite 4, LPM4 (Low Power Mode 4).

Le mode LPM4 est celui qui permet le plus de conserver de l'énergie. Nous avons réalisé plusieurs expérimentation en se basant sur ces différents modes afin d'allonger la durée de vie des capteurs. Nous détaillerons ces expérimentations dans le chapitre contribution.

Pour activer les capteurs (accéléromètre et luminosité) présents dans le Nanosensor N741, il faudra activer ou désactiver une ou plusieurs entrées/sorties du convertisseur analogique numérique (cf. figue 22). Ces manipulations sont traduites en langage C que nous compilons et envoyons sur les Nanosensors à travers l'application BSL Scriptor en utilisant le câble USB.

```

////////////////////////////////////
// illumination sensor P74 enable & P75 adc in
////////////////////////////////////
P7DIR |= 0x10;
P7OUT |= 0x10;
P7REN &= 0xdf; // no pull up resistor

////////////////////////////////////
// acc sensor P40 enable and Gsel P41, 440mV/G if 3G mode else 118mV/G, 0=3G, 1=11G mode
////////////////////////////////////
P4DIR |= 0x03;
P4OUT |= 0x01; // enable to 1 and sensitivity to 3G mode
// ADC setup
P6SEL |= 0x17; // Enable A/D 0-2 for acc. sensor and 4 for
P7SEL |= 0x08; // Enable A/D channel 13 = P75

```

Figure 22 Activation des capteurs de luminosité et d'accélération

3.2.2 L'unité de transmission

Il s'agit du module Radiocrafts RC2400HP. Ce module est composé d'un SoC (System On Chip) CC2530 et d'un processeur 8051 de Texas Instrument. Il a été spécialement conçu pour les réseaux utilisant le standard IEEE 802.15.4. Les caractéristiques de ce module sont illustrées dans le tableau 1.

Tableau 1 Caractéristiques du module de transmission [21]

Parameter	RC2400(HP)-ZNM	Unit
Frequency band	2.400-2.4835	GHz
Number of channels	16	
Data rate	250	kbit/s
Max output power	4/20*	dBm
Sensitivity (PER 1%)	-97/-99*	dBm
Adjacent Channel Rejection	35	dB
Alternate Channel Rejection	56	dB
Supply voltage	2.0/2.7*-3.6	Volt
Current consumption, RX/TX	27	mA
Current consumption, PD	0.6	uA
Flash Memory	256	kB
EEPROM (optional)	4	kB
Operating Temperature	-30 to +85	C

Le nombre de canaux radio fréquences utilisables est égale à onze. Chaque canal est d'une largeur de 5 MHz (Cf. Tableau 2). Pour que les capteurs soient dans le même réseau, il faut qu'ils utilisent tous le même canal. Dans notre cas, nous avons utilisé le canal 11 (2405 MHz).

Tableau 2 Canaux Radio Fréquences utilisés par RC2400HP

RF channel	Frequency
11	2405 MHz
12	2410 MHz
13	2415 MHz
14	2420 MHz
15	2425 MHz
16	2430 MHz
17	2435 MHz
18	2440 MHz
19	2445 MHz
20	2450 MHz
21	2455 MHz
22	2460 MHz
23	2465 MHz
24	2470 MHz
25	2475 MHz
26	2480 MHz

Nous pouvons programmer la puissance du signal de l'unité de transmission afin de gagner en énergie dans le cas d'un réseau à forte densité ou pour avoir une couverture réseau maximale dans le cas d'un réseau à faible densité et où la distance entre les capteurs est importante.

Nous présentons dans le tableau 3 les valeurs possibles et nous verrons dans la section qualité de service du chapitre contributions les expérimentations faites à ce niveau.

Tableau 3 Gestion de la puissance du module RC2400HP

Power setting (TXPOWER)	Output power (dBm)	Peak current (mA)
0xF5	20,3	218
0xE5	19,5	192
0xD5	18,8	174
0xC5	18,1	160
0xB5	17,2	145
0xA5	16,2	134
0x95	14,5	121
0x85	13,5	112
0x75	12,0	99
0x65	10,5	93

La plus importante fonctionnalité de l'unité de transmission RC2400HP est la possibilité d'avoir en plus des modes émission et réception, trois modes en basse consommation d'énergie (cf. figure 23). Pour chacun de ces modes, la consommation d'énergie est différente, ceci nous aidera dans l'élaboration d'un plan de conservation d'énergie.

Low Power

- **Active-Mode RX (CPU Idle): 24 mA**
- **Active Mode TX at 1 dBm (CPU Idle): 29 mA**
- **Power Mode 1 (4 μ s Wake-Up): 0.2 mA**
- **Power Mode 2 (Sleep Timer Running): 1 μ A**
- **Power Mode 3 (External Interrupts): 0.4 μ A**

Figure 23 Modes de consommation d'énergie de l'unité de transmission

Dans cette section nous avons présenté la partie matérielle du kit 6LoWPAN de Sensinode. Dans la section qui va suivre, nous allons présenter l'application NodeView 2.0.

3.3 NodeView 2.0

3.3.1 Présentation

NodeView 2.0 est une application développée en Java. Elle est fournie par Sensinode, elle présente plusieurs fonctionnalités auxquelles nous avons apporté des modifications pour mieux l'adapter à nos besoins. L'interface comporte quatre onglets principaux qui sont :

- **Router view** : contient toutes les informations relatives aux routeurs connectés comme le nom, l'adresse IPv6, la date de démarrage, le nombre de capteurs avec qui un lien est établi et les informations. On y trouve aussi les mêmes informations concernant les capteurs ;
- **Message log** : contient les données reçues en format hexadécimale plus l'heure d'arrivée, la provenance et le port de destination ;

- **Network analyzer** : on y trouve l'architecture du réseau ainsi que des informations comme le débit, le nombre de paquets émis et reçus, etc. ;
- **NanosensorTab** : c'est là où seront affichées les données reçues ainsi que les métadonnées comme l'adresse mac, l'adresse IP et la référence du capteur.

3.3.2 Fonctionnement

Une fois l'application démarrée (cf. figure 24) et la communication établie avec le Nanorouter, nous pouvons visualiser les nœuds connectés, leurs adresses IPv6 et MAC, le nombre de paquets émis, etc.

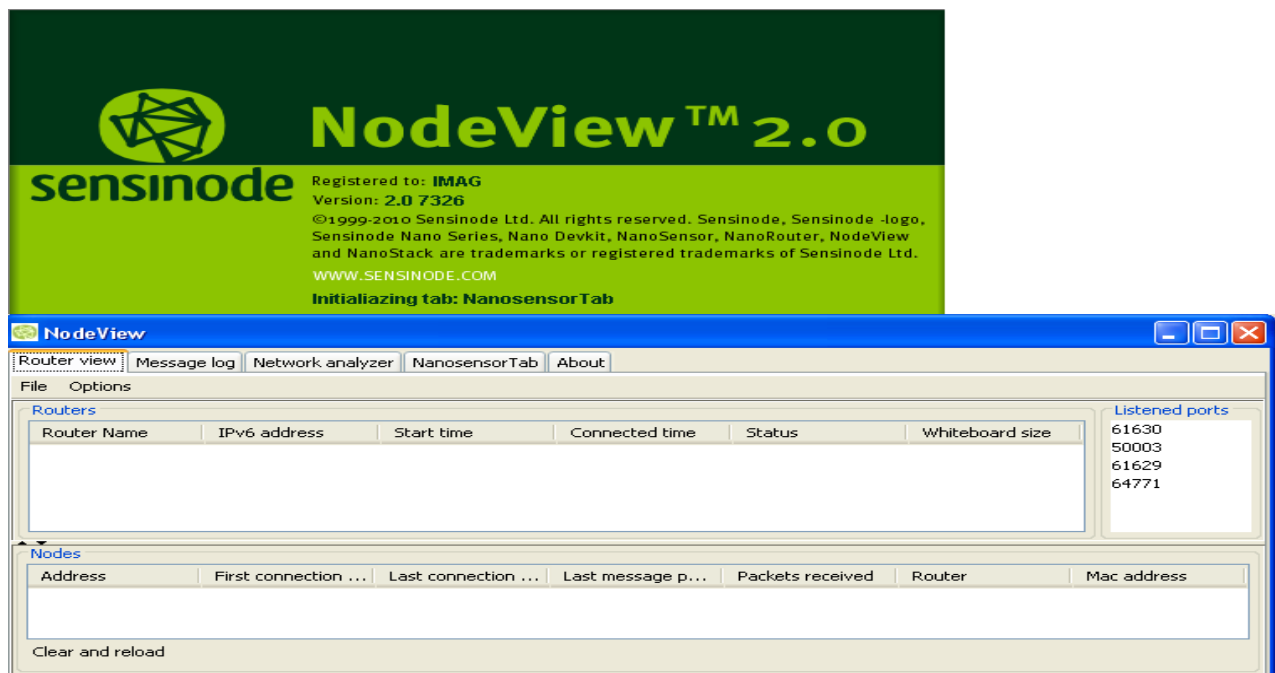


Figure 24 Démarrage de NodeView 2.0

Pour commencer à utiliser NodeView 2.0, il suffit d'avoir l'adresse Ipv6 du Nanorouter pour pouvoir le connecter (cf. figure 25) et pouvoir par la suite administrer notre réseau de capteurs.

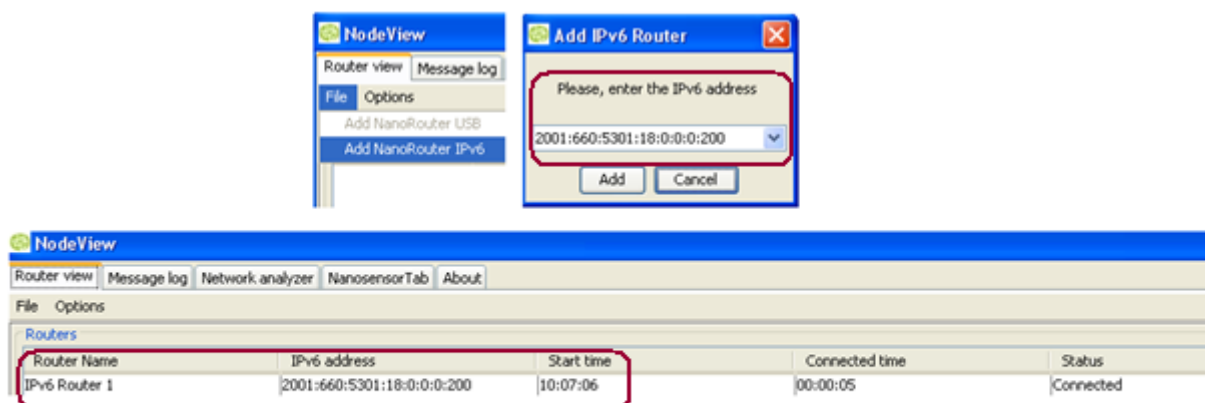


Figure 25 Connecter le Nanorouter à NodeView 2.0

Du moment où des capteurs sont démarrés, ils seront affichés dans l'interface principale de NodeView 2.0 (cf. figure 26). En effet, le Nanorouter après avoir établi la communication avec les nœuds détectés en utilisant la technologie 6LoWPAN et son protocole de routage RPL, ouvre une communication en UDP sur le port 64771 avec ces derniers. Les capteurs sont aussi programmés de la même façon. NodeView 2.0 pourra à ce moment recevoir les données en provenance des capteurs à travers le Nanorouter.

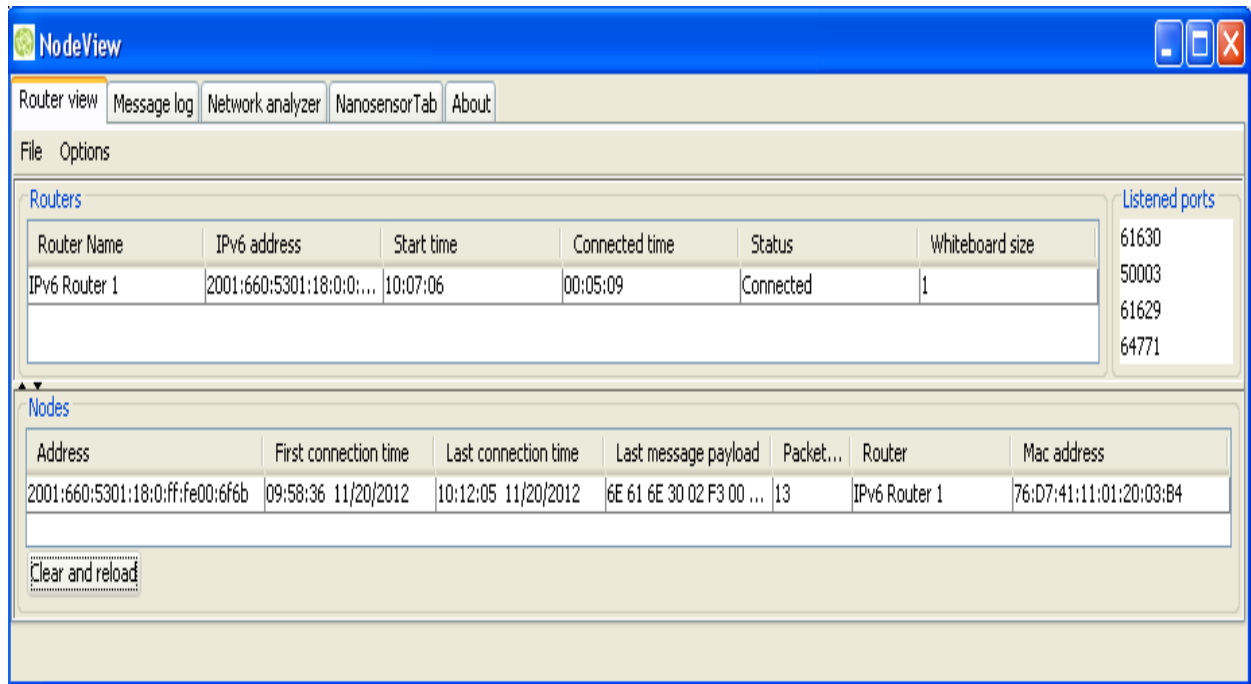


Figure 26 Capteur connecté à NodeView 2.0

Nous avons deux types de données (cf. figure 27) en plus des données standards comme les adresses IPv6 et MAC, la date de réception du paquet, le nombre de paquets reçus, etc.

NodeView			
Router view Message log Network analyzer NanosensorTab About			
Time stamp	Connection from	Destination port	Data
09:58:36:437	2001:660:5301:18:0:ff:fe00:6f6b	61630	6E 52 54 11 09 C8 6F 68 00 AA 76 01 00 00 00 00 00 76 D7 41 11 01 20 03 B4 00 00 00 00 00
09:58:36:437	2001:660:5301:18:0:ff:fe00:6f6b	61629	6E 61 6E 30 02 F3 00 05 19 4E 37 34 31 20 4D 41 43 3A 37 36 44 37 34 31 31 31 30 31 32 30 30 33 42 34 F3 01 02 08 4D
10:10:40:546	2001:660:5301:18:0:ff:fe00:6f6b	61629	6E 61 6E 30 02 F3 00 05 19 4E 37 34 31 20 4D 41 43 3A 37 36 44 37 34 31 31 31 30 31 32 30 30 33 42 34 F3 01 02 08 2F
10:10:51:187	2001:660:5301:18:0:ff:fe00:6f6b	61629	6E 61 6E 30 02 F3 00 05 19 4E 37 34 31 20 4D 41 43 3A 37 36 44 37 34 31 31 31 30 31 32 30 30 33 42 34 F3 01 02 08 5C
10:11:01:859	2001:660:5301:18:0:ff:fe00:6f6b	61629	6E 61 6E 30 02 F3 00 05 19 4E 37 34 31 20 4D 41 43 3A 37 36 44 37 34 31 31 31 30 31 32 30 30 33 42 34 F3 01 02 08 3E
10:10:37:046	2001:660:5301:18:0:ff:fe00:6f6b	61630	6E 52 54 11 09 C8 6F 68 00 AA 76 01 00 00 00 00 00 76 D7 41 11 01 20 03 B4 00 00 00 00 00

Figure 27 Types de données reçues

Les premiers concernent la topologie du réseau, ils sont envoyés sur le port udp 61630 et sont traduits dans l'onglet *Network analyzer* comme le montre la figure 28.

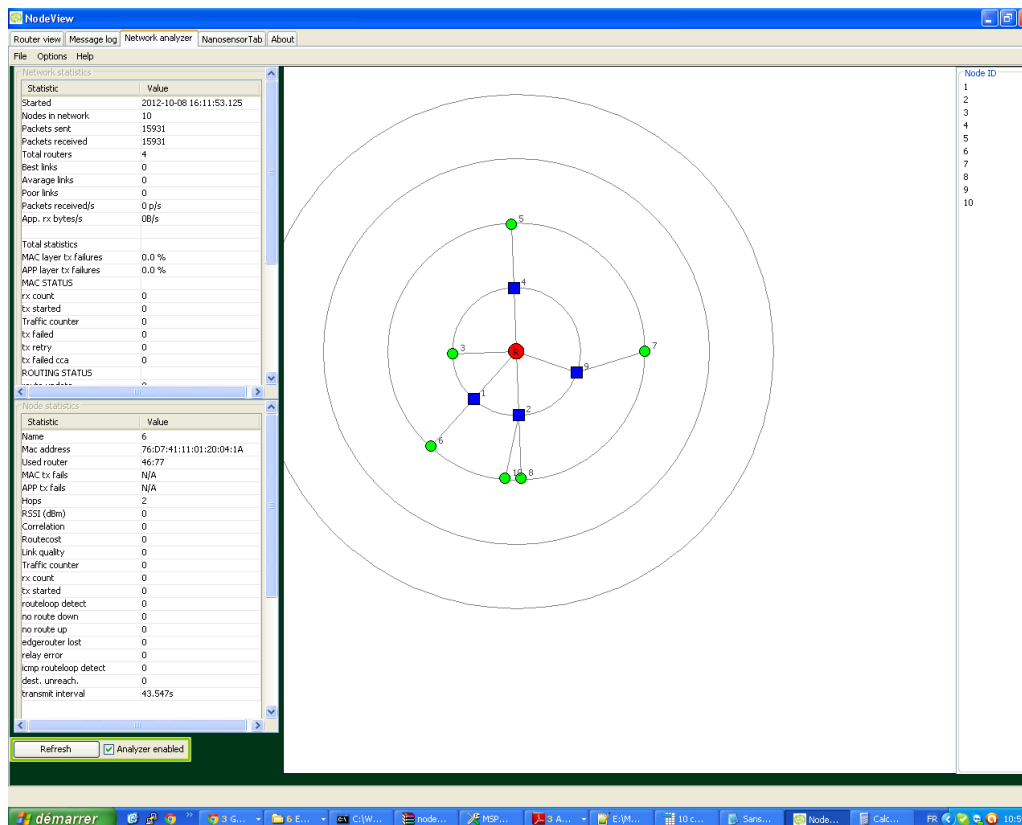


Figure 28 L'onglet Network analyzer

Les deuxièmes concernent les mesures relevées. Les mesures sont envoyées sur le port udp 61629 et sont affichées dans l'onglet NanosensorTab comme le montre la figure 29.

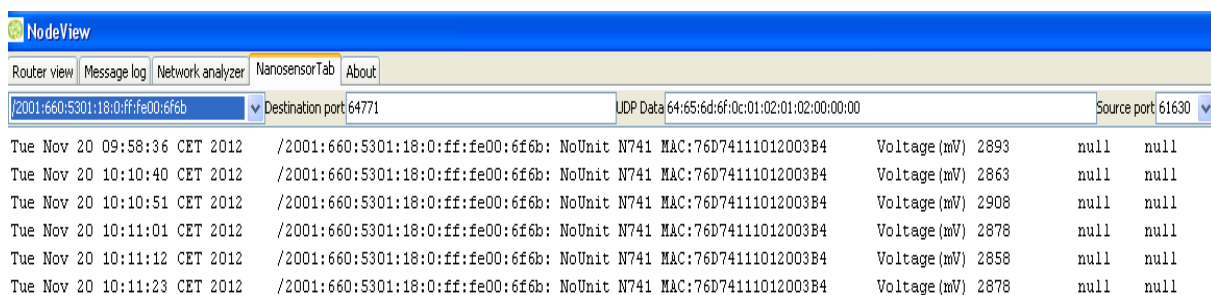


Figure 29 L'onglet NanosensorTab

Le logiciel fourni par Sensinode offre plusieurs fonctionnalités intéressantes pour gérer le matériel acquis par l'équipe SIGMA. Néanmoins, des modifications lui ont été apportées afin qu'il réponde à nos besoins notamment en terme de gestion de données.

3.3.3 Modification de NodeView 2.0

L'aspect gestion des données est une des tâches que je devais accomplir pendant ce projet, c'est pour cela que les modifications apportées concernent l'onglet NanosensorTab de NodeView 2.0. Il y avait la possibilité de rajouter de nouveaux onglets mais pour nous l'idée était d'améliorer l'existant. Nous présenterons plus en détail notre apport dans le chapitre contributions, au niveau de la section gestion des données.

4 Conclusion

L'internet des objets et les réseaux intelligents sont l'avenir de l'internet, c'est pour cela que Sensinode a regroupé plusieurs technologies qui vont dans ce sens dans le développement de son kit 6LoWPAN. Le matériel fourni répond à certaines caractéristiques de l'internet des objets notamment la mobilité, l'auto-configuration, etc. mais des solutions qui répondent à la problématique de la consommation d'énergie et la qualité de service sont nécessaires. Le microcontrôleur MSP430F5438 qui représente les unités d'acquisition et de traitement des Nanosensors est à étudier et explorer pour pouvoir mettre en marche les capteurs de luminosité et d'accélération qui l'équipent.

Dans ce chapitre, nous avons présenté le matériel Sensinode 6LoWPAN que l'équipe SIGMA s'est procuré. Nous avons présenté les technologies utilisés, les composants ainsi que l'application NodeView 2.0.

Dans le chapitre suivant, nous allons présenter nos différentes contributions dans ce projet. Les contributions concernant la gestion des données seront présentées dans la section 1, la section 2 contiendra différentes idées proposées et expérimentations réalisées afin de répondre aux besoins en termes de qualité de service, enfin nous présenterons dans la section 3 les techniques et simulations que nous avons utilisées pour limiter au maximum la consommation de l'énergie.

Chapitre 3 : Contributions

Les applications à base de réseaux de capteurs demandent à ces dispositifs de leur fournir des données fiables et exploitables pour satisfaire les besoins des utilisateurs. Il faut programmer les capteurs de telle sorte de pouvoir générer des données pertinentes. Pour arriver à ce but, il faudra faire face aux différentes contraintes que connaît ce genre de réseaux notamment en terme de support de la qualité de service en particulier la tolérance aux pannes et de consommation d'énergie.

Dans ce chapitre, nous allons présenter nos contributions dans le but de répondre aux objectifs qui étaient fixés :

- Programmer les capteurs pour les mesures, l'échantillonnage et l'envoi des paquets ;
- Adapter l'application NodeView 2.0 fournie par Sensinode à nos besoins en terme de gestion des données et communication avec les capteurs ;
- Proposer des solutions pour le support de la qualité de service ;
- Etudier le comportement des capteurs en termes de consommation d'énergie et proposer des idées et solutions afin de réduire au maximum cette consommation.

D'une façon générale, la gestion de notre s'est opérée suivant une démarche itérative selon la roue de Deming, illustrée sur la figure 30, en suivant sur chaque itération les quatre étapes suivantes (en anglais, Plan - Do - Check - Act) :

- planification : on planifie les tâches à accomplir sur cette itération ;
- action : on tente d'accomplir les tâches définies précédemment dans le délai imparti ;
- vérification : on vérifie si le travail réalisé correspond aux besoins exprimés ;
- ajustement : on recherche les améliorations à apporter au sujet.

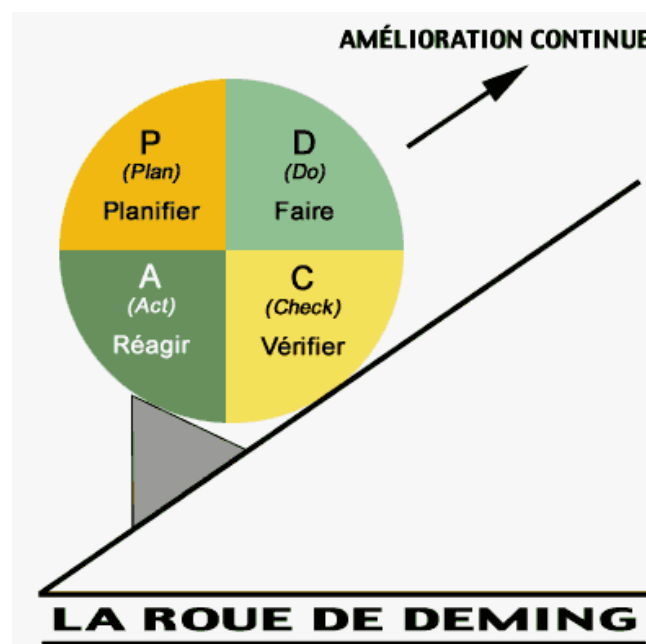


Figure 30 Les quatre étapes d'une démarche itérative selon la roue de Deming

Selon la tâche à réaliser, un cycle de la roue peut varier d'une semaine à un mois à l'issue duquel une réunion avec mon tuteur, Cyril Labbé, est organisée. Lors de cette réunion, nous faisons le point sur les tâches réalisées (check) et nous établissons les priorités pour le cycle suivant (plan).

Nos contributions visent trois points. Le premier point concerne la gestion des données, il sera présenté dans la première section. La deuxième section portera sur nos travaux autour du support pour la qualité de service, les métriques fixées et les expérimentations réalisées. Dans la section 3, nous présenterons le comportement des Nanosensors au niveau de la consommation d'énergie, les propositions et expérimentations effectuées pour rallonger au maximum la durée de vie du réseau Sensinode. Nous finirons par une conclusion.

1 Gestion des données

Nos contributions concernant l'aspect gestion des données se divisent en deux parties. La première partie porte sur l'exploitation du microcontrôleur³ qui représente l'unité d'acquisition et l'unité de traitement des données des Nanosensors. La deuxième partie porte sur les modifications apportées à NodeView 2.0.

1.1 Programmation des Nanosensors

La programmation du microcontrôleur passe par quatre étapes :

- **Etape1** : la lecture et la compréhension de la documentation du constructeur Texas Instrument. Il s'agit des documents *MSP430x5xx/MSP430x6xx Family User's Guide* et *MSP430F543x, MSP430F541x MIXED SIGNAL MICROCONTROLLER*. Le but étant de se familiariser avec ce type de composants électroniques et de pouvoir exploiter leurs fonctionnalités.
- **Etape 2** : elle consiste à traduire les besoins en langage C ;
- **Etape 3** : compiler le code en utilisant le logiciel IAR Embedded Workbench IDE qui est le compilateur conseillé par Sensinode et Texas Instrument. La version utilisée est la version IAR Embedded Workbench for MSP430 5.40.6 ;
- **Etape 4** : transférer le code compilé sur le microcontrôleur.

1.1.1 Activation des capteurs

Nous avons trois unités de mesures qui peuvent être activées sur le MSP430F5438, il s'agit des capteurs d'accélération, de niveau de batterie et de luminosité. Le projet fourni par Sensinode que nous avons nommé *MSP430* comporte un exemple de code qui permet de programmer les capteurs. Ce dernier nous a beaucoup aidé, en effet les deux premiers capteurs étaient déjà opérationnels. Il fallait donc suivre l'exemple pour pouvoir mettre en marche le capteur de luminosité. Les instructions relatives à l'activation des capteurs et la gestion des signaux électriques sont introduites dans le fichier *adc_driver.c*.

³ MSP430F5438 Ultra Low-Power Microcontroller

```

////////////////////////////////////////
// illumination sensor P74 enable & P75 adc in
////////////////////////////////////////
P7DIR |= 0x10;
P7OUT |= 0x10;
P7REN &= 0xdf; // no pull up resistor

P7SEL |= 0x08; // Enable A/D channel A13 = P75

ADC12MCTL0 = ADC12SREF_1 + ADC12EOS + ADC12INCH_13;

```

Figure 31 Activation du capteur de luminosité sur le MSP430F5438

Comme le montre la figure 31, l'activation du capteur de luminosité consiste à sélectionner le canal 13 du port registre P7 (P7SEL|= 0x08) puis indiquer au convertisseur analogique numérique (ADC) la direction du signal électrique, c'est-à-dire en sortie (P7DIR|= 0x10 et P7OUT|= 0x10). Après, la valeur de sortie sera redirigée vers le registre ADC12MCTL0.

L'étape suivante consiste à démarrer l'ADC (ADC12CTL0 = ADC12ON), activer la conversion au niveau de l'ADC (ADC12CTL0|= ADC12ENC) et la démarrer (ADC12CTL0|= ADC12SC) pour que ce dernier puisse convertir les signaux électriques en données numériques (cf. figure 32).

```

ADC12CTL0 = ADC12ON+ADC12SHT02+ADC12REFON+ADC12REF2_5V;
// Turn on ADC12, Sampling time
ADC12CTL0 |= ADC12ENC; // Enable conversions
ADC12CTL0 |= ADC12SC; // Start conversion

```

Figure 32 Le convertisseur Analogique Numérique ADC12

Pour chaque capteur, un canal de l'ADC est affecté, pour le capteur de luminosité il s'agit de ADC12INCH_13. Cette valeur est ensuite stockée dans le registre ADC12MCTL0 (cf. figure 33) et puis dans la mémoire ADC12MEM0. C'est la valeur contenue dans ce registre, à travers les fonctions intermédiaires (cf. figure 34) qui va être envoyée à l'application NodeView 2.0.

```

uint8_t read_AD(uint8_t channel)
{
    AD_ready = 0;
    ADC12CTL0 &= 0xffffd;
    switch(channel)
    {
        case 0x0d:
            ADC12MCTL0 = ADC12INCH_13;
            break; //illumination sensor
    }
}

```

Figure 33 ADC12INCH13

```

uint8_t AD_read_ill_data(void)
{
    read_AD(0x0d);
    while(!AD_ready);
    ad_measure_values.light = AD_value;

    return 1;
}

```

Figure 34 Lecture de la valeur luminosité

1.1.2 Création des messages

La création des messages est définie par la fonction *create_nanossoap_msg* dans le fichier *nanossoap.c*. Dans cette fonction, nous avons défini un buffer pour gérer la file d'attente,

ensuite selon le type de données (niveau de batterie, accélération ou luminosité), un message est créé et envoyé au buffer. Chaque message est composé de plusieurs champs, ceci dépend principalement du type de la mesure à envoyer. Par exemple, si le message concerne l'accélération, nous aurons trois champs correspondant chacun à un axe de mesure (accX, accY et accZ).

Ensuite, la valeur `ad_measure_value.xxx` envoyée par la fonction `AD_read_xxx_data` sera envoyée au buffer. La figure 35 illustre les fonctions utilisées pour la création des messages.

```
uint8_t create_nanosoaap_msg(uint8_t *buffer, uint8_t *data, uint8_t dlength, uint8_t messagetype)
{
    uint8_t *dptr= buffer;
    switch(messagetype)
    {
        case 0x01: *dptr++ = 0x02; break;          /* 2 Tags in array, nodetype(String) and battery voltage */
        case 0x02: *dptr++ = 0x01; break;          /* illumination sensor */
        // case 0x03: *dptr++ = 0x03; break;          /* 3 Tags in array, acceleration x,y,z */
        default: *dptr++ = 0x01; break;
    }
    .
    .
    .
    else if(messagetype == 0x02) /* illumination */
    {
        *dptr++ = 17;          /* ill*/
        *dptr++ = 0x02;          /* 16-bit variable */
        *dptr++ = (uint8_t)(ad_measure_values.light >> 8);
        *dptr++ = (uint8_t) ad_measure_values.light;
        /**dptr++ = 0xf3;          /* MeasurementElement */
    }
    .
    .
    .
    return dptr - buffer;    /*send_data_to_router_address(napmsg, dptr, 61629, 61629) ;
}
```

Figure 35 Création des messages

1.1.3 Envoi des messages

C'est dans le fichier *main.c* que l'appel aux différentes fonctions décrites précédemment est effectué. Chaque capteur est associé à un *timer*, nous en avons trois, un pour chaque mesure relevée : BATTERY_INFO pour le niveau de batterie, ACC_INFO pour l'accéléromètre et ILL_INFO pour la luminosité. Pour chacun des cas, plusieurs conditions sont à vérifier avant de pouvoir envoyer le message.

- **Appel de la fonction** qui lit la valeur stockée dans la mémoire : `AD_read_battery_data ()` pour les piles, `AD_read_acc_data ()` pour le capteur d'accélération et `AD_read_ill_data ()` pour le capteur de luminosité ;
- **Vérification du message** : il s'agit de vérifier si le message a bien été créé par la fonction `create_nanosoaap_msg()` ;
- **Spécification du numéro de port** : c'est le port UDP 61629 qui est donné, une socket est ouverte entre NodeView et le Nanorouter utilisant ce port ;
- **Faire appel à la socket d'envoi** : dans cette socket, nous spécifions son identifiant, l'adresse de NodeView et le contenu du buffer ;
- **Faire appel à la fonction `timer_sys_event()`** : dans cette fonction nous désignons la mesure à envoyer et la fréquence d'envoi des messages.

Un exemple d'envoi de message des mesures de luminosité est illustré dans la figure 36. Dans cet exemple, on spécifie une fréquence d'envoi de 30 secondes (30000 millisecondes).

```

case ILL_INFO:
{
    uint8_t msg_data[3];
    AD_read_ill_data();

    //printf("ADC reading illumination\r\n");    //, moh comment

    if(boolean_analyzer_address_known)
    {
        uint8_t msg_data[10];
        if(nanosoap_len = create_nanosoap_msg(tx_buffer,msg_data, 0, 0x02))
        {
            if (er_state != 0)
            {
                nodeview_address.port = 61629;
                socket_sendto(socket_id,nodeview_address,tx_buffer,nanosoap_len);
            }
        }
    }

    timer_sys_event_cancel(ILL_INFO,1);
    timer_sys_event((uint8_t)ILL_INFO,30000,1);
}

break;

```

Envoi des messages
toutes les 30 secondes

Figure 36 Envoi des mesures

Les différentes étapes que nous venons de présenter sont essentielles dans le cheminement d'une mesure. Nous avons illustré dans la figure 37 le cycle de vie d'une mesure dès l'activation du capteur jusqu'à son envoi à NodeView.

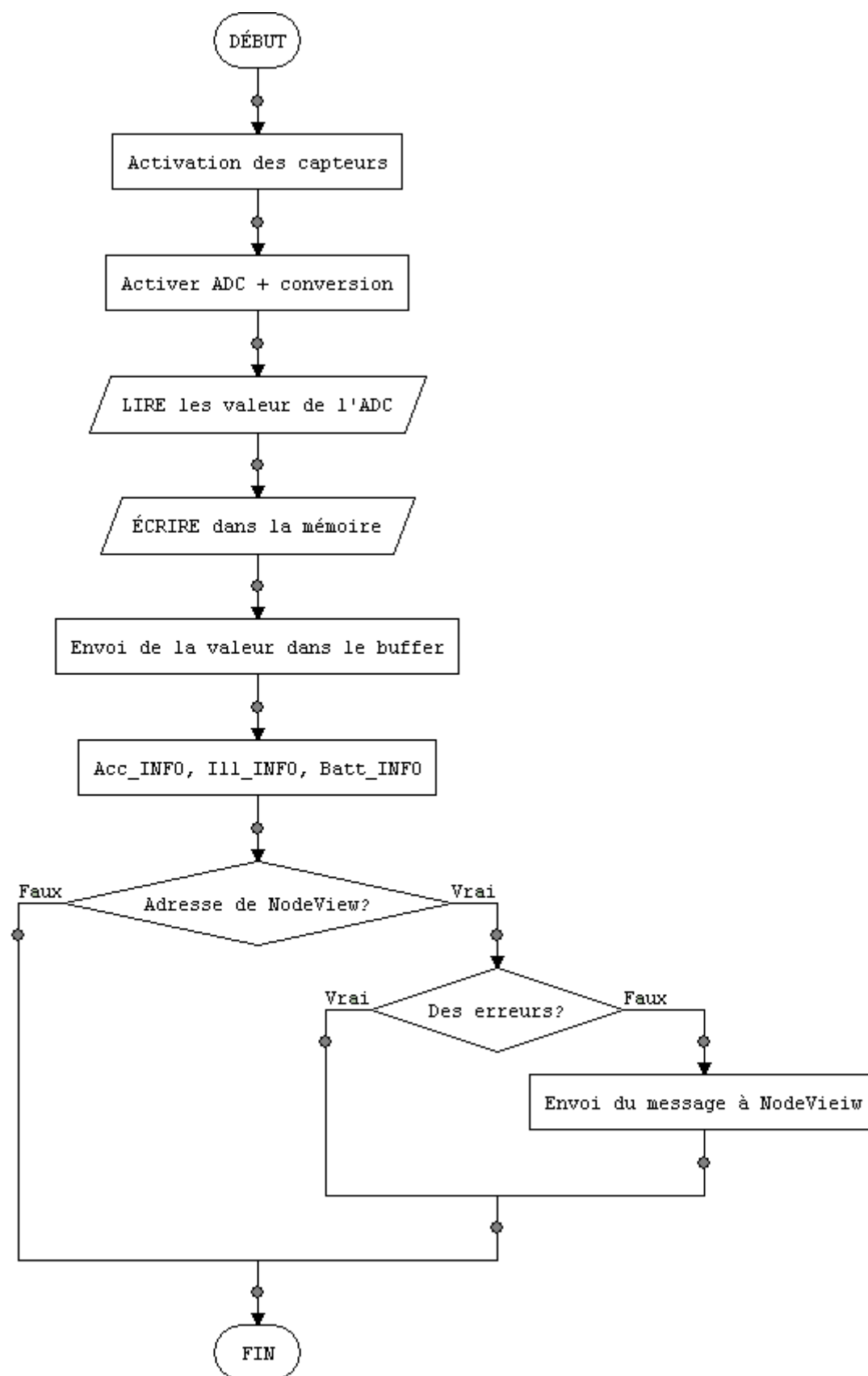


Figure 37 Cycle de vie d'une mesure

1.1.4 Ajout de l'opérateur de sélection

Afin de limiter l'envoi des mesures redondantes en provenance d'un même Nanosensor, nous avons défini un opérateur de sélection au niveau du microcontrôleur MSP430F5438. La sélection des mesures aide aussi à conserver l'énergie des capteurs puisque c'est l'envoi des messages qui coûte le plus cher dans la consommation d'énergie. Au bout de plusieurs tentatives, nous avons réussi à mettre en place cette technique et la développer par la suite. Le principe est de définir un seuil, si la valeur relevée par le capteur est inférieure à ce seuil, il n'y aura pas de message envoyé, si la valeur est supérieure à ce seuil, le message est créé et envoyé à NodeView 2.0. Quatre types de sélection ont été mises en place :

Sélection simple : dans ce cas, nous ne nous intéressons qu'à un seul capteur par exemple celui de la luminosité et une seule condition est à vérifier. La figure 38 montre un exemple de cette opération.

```
case ILL_INFO:
{
    uint8_t msg_data[3];
    AD_read_ill_data();
    if (ad_measure_values.light > 10)
    {
        timer_sys_event_cancel(ILL_INFO,1);
        timer_sys_event((uint8_t)ILL_INFO,10000,1);
    }
}
```

Seuil = 10

Figure 38 Opérateur de sélection simple

Sélection multiple : dans ce cas, nous avons défini des paliers. C'est-à-dire, si la mesure est supérieure au seuil 1, le message est envoyé toutes les xx secondes, si la mesure est supérieure au seuil 2, le message est envoyé toutes les yy secondes et ainsi de suite. Un exemple est donné dans la figure 39.

```
case ILL_INFO:
{
    uint8_t msg_data[3];
    AD_read_ill_data();
    if (ad_measure_values.light > 10)
    {
        timer_sys_event_cancel(ILL_INFO,1);
        timer_sys_event((uint8_t)ILL_INFO,10000,1);
    }
    if (ad_measure_values.light > 20)
    {
        timer_sys_event_cancel(ILL_INFO,1);
        timer_sys_event((uint8_t)ILL_INFO,15000,1);
    }
}
```

Seuil 1

Seuil 2

Figure 39 Opérateur de sélection multiple

Sélection Combinée simple : il s'agit de combiner deux mesures provenant de deux sources différentes avec un seuil pour chaque mesure : par exemple, l'unité de mesure du niveau de batterie et celle de la luminosité. Cette technique est présentée dans la figure 40.

```
case ILL_INFO:
{
    uint8_t msg_data[3];
    AD_read_ill_data();

    if (ad_measure_values.light > 10) && ad_measure_values.battery > 2500)

        timer_sys_event_cancel(ILL_INFO,1);
        timer_sys_event((uint8_t)ILL_INFO,10000,1);
}
```

Seuil luminosité

Seuil batterie

Figure 40 Opérateur de sélection combinée simple

Sélection Combinée multiple : cette technique combine les trois précédentes. Nous prenons l'exemple de luminosité pour expliquer la démarche. Plusieurs seuils correspondants à différents paliers ont été définis. Pour chaque palier, la valeur du seuil défini pour la luminosité est combinée avec un seuil de l'information envoyée par l'unité de mesure de la batterie et un *timer* est associé à chaque niveau.

Avec cette technique, nous pouvons optimiser l'envoi des données par les Nanosensors. Nous avons d'une part une limitation d'envoi des mesures redondantes et d'autre part la possibilité de surveiller le niveau de la batterie par déduction.

La figure 41 illustre la technique d'instauration d'un opérateur de sélection combiné multiple.

```
case ILL_INFO:
{
    uint8_t msg_data[3];
    AD_read_ill_data();

    if (ad_measure_values.light > 10 && ad_measure_values.battery > 2500) // opérateur de sélection
    {
        timer_sys_event_cancel(ILL_INFO,1);
        timer_sys_event((uint8_t)ILL_INFO,10000,1);
    }

    if (ad_measure_values.light > 20 && ad_measure_values.battery < 2500 && ad_measure_values.battery > 2300)
    {
        timer_sys_event_cancel(ILL_INFO,1);
        timer_sys_event((uint8_t)ILL_INFO,20000,1);
    }

    if (ad_measure_values.light > 30 && ad_measure_values.battery > 2000 && ad_measure_values.battery < 2300)
    {
        timer_sys_event_cancel(ILL_INFO,1);
        timer_sys_event((uint8_t)ILL_INFO,30000,1);
    }
}

break;
```

Seuils luminosité

Seuils batterie

timer 1

timer 2

timer 3

Figure 41 Sélection combinée multiple

Les opérateurs de sélection ajoutés permettent de gérer les données mais ont leur impact sur la qualité de service. En effet, nous arrivons à mieux conserver l'énergie parce qu'on envoie plus ou moins de messages mais ceci peut impacter la qualité des données et la couverture temporelle.

Après avoir développé les différentes fonctions, compilé le code et corrigé les bugs, l'étape suivante était de flasher le microcontrôleur MSP430F5438.

1.1.5 Flash du microcontrôleur MSP430F5438

Cette opération est réalisée à l'aide du programme BSL Scriptor qui est un outil spécifique pour la famille MSP430 et qui s'exécute en ligne de commande. Le programme appelle un fichier texte que nous avons appelé *script.txt* dans lequel nous avons précisé les opérations à effectuer. Il faut spécifier principalement le type du microcontrôleur, le port COM utilisé et surtout le nom du fichier contenant le nouveau firmware (cf. figure 42). Ce dernier contient un ensemble d'instructions écrites en hexadécimale.

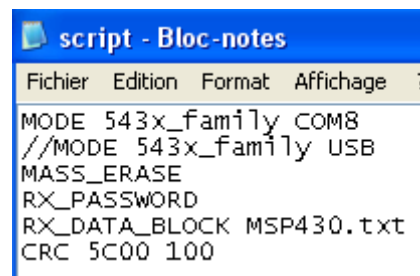


Figure 42 Le fichier script.txt

L'opération de mise à jour du firmware est illustrée dans la figure 43.

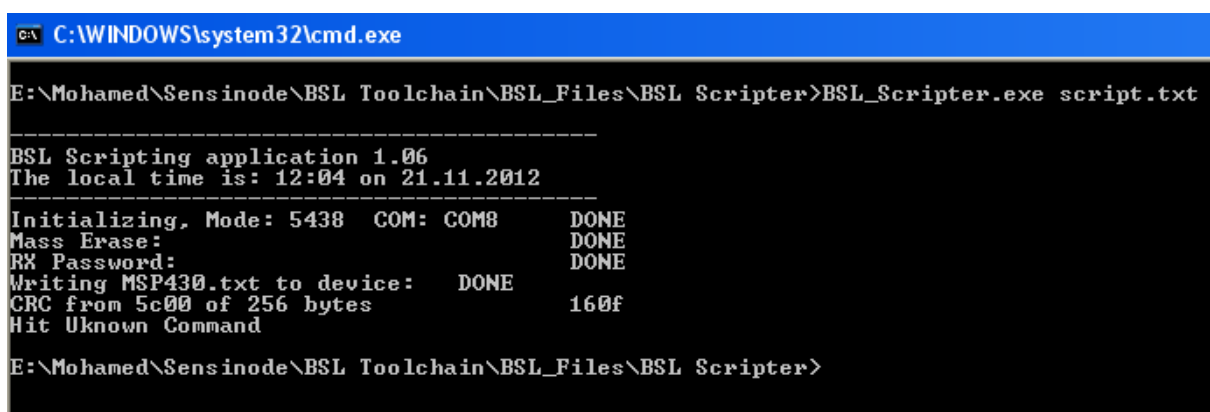


Figure 43 Mise à jour du firmware des Nanosensors

La programmation des Nanosensors demande une maîtrise du fonctionnement du microcontrôleur MSP430F5438 et de la programmation embarquée. Nous avons pu développer les fonctionnalités de nos capteurs et en ajouter d'autres comme l'opérateur de sélection qui n'était pas présent sur le matériel Sensinode.

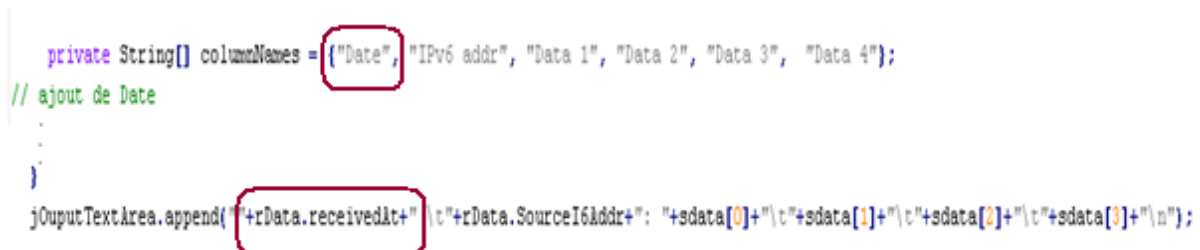
Dans la section suivante nous allons présenter nos travaux au niveau de l'application NodeView 2.0.

1.2 Amélioration de NodeView 2.0

NodeView 2.0 a été livré avec quelques fichiers sources que nous pouvions modifier, compiler et intégrer dans l'exécutable. Les modifications concernent principalement l'onglet NanosensorTab dans lequel seront affichées les données envoyées par les Nanosensors.

1.2.1 Ajout de la colonne date de réception

Il s'agit d'afficher pour chaque donnée reçue la date et l'heure correspondante. En effet dans l'état actuel de l'application, nous pouvons voir dans l'onglet NanosensorTab l'adresse IP des capteurs, son adresse mac, le type de nœud et la valeur mesurée. En ajoutant la date et l'heure de réception, nous pourrions vérifier s'il y a eu une perte de paquets et déterminer les délais de transmission. Ces métriques sont fondamentales dans la mesure de la qualité de service. Cette modification est présentée dans la figure 44.



```
private String[] columnNames = {"Date", "IPv6 addr", "Data 1", "Data 2", "Data 3", "Data 4"};
// ajout de Date
.
.
}

jOutputTextArea.append("\rData.receivedAt:" + rData.receivedAt + "\t" + rData.SourceIPv6Addr + ":" + sdata[0] + "\t" + sdata[1] + "\t" + sdata[2] + "\t" + sdata[3] + "\n");
```

Figure 44 Ajout de la colonne Date de réception

1.2.2 Ajout de la colonne Modèle du capteur

Sensinode propose plusieurs modèles de capteurs. Nous en connaissons trois qui sont N711, N741 et N751. Cette fonctionnalité nouvelle permet d'afficher le modèle du capteur qui a envoyé la donnée. Nous en voyons l'utilité dans le cas où plusieurs modèles de capteurs Sensinode sont utilisés dans un même réseau.

1.2.3 Disposition des colonnes

Pour avoir une meilleure ergonomie de l'interface et faciliter la lecture des données reçues, nous avons modifié le fichier NanosensorTab.java pour arriver à avoir les colonnes dans l'ordre suivant :

- Date de réception : il s'agit de la date de réception de la mesure. Ce champ contient la date et l'heure ;
- L'adresse IP du Nanosensor : c'est l'adresse IPv6, elle est récupérée à partir du fichier *rData.java*. Cela permet de ne pas confondre les provenances des données ;
- L'adresse MAC : elle permet de filtrer les données en provenance d'un même capteur mais qui a changé plusieurs fois d'adresse IP ;
- Le type de mesure : dans le cas où nous recevons différentes mesures d'un même Nanosensor par exemple le niveau de batterie, l'accélération et la luminosité, ceci permettra de faire la distinction ;
- La valeur de la mesure : il s'agit de la valeur numérique reçue.

L'onglet NanosensorTab modifié est présenté dans la figure 45.

Date et heure	Adresse IPv6	Le modèle	Adresse MAC	Type de mesure	Valeur mesurée
Tue Nov 20 09:58:36 CET 2012	/2001:660:5301:18:0:ff:fe00:6f6b	NoUnit N741	MAC:76D74111012003B4	Voltage (mV)	2893
Tue Nov 20 10:10:40 CET 2012	/2001:660:5301:18:0:ff:fe00:6f6b	NoUnit N741	MAC:76D74111012003B4	Voltage (mV)	2863
Tue Nov 20 10:10:51 CET 2012	/2001:660:5301:18:0:ff:fe00:6f6b	NoUnit N741	MAC:76D74111012003B4	Voltage (mV)	2908
Tue Nov 20 10:11:01 CET 2012	/2001:660:5301:18:0:ff:fe00:6f6b	NoUnit N741	MAC:76D74111012003B4	Voltage (mV)	2878
Tue Nov 20 10:11:12 CET 2012	/2001:660:5301:18:0:ff:fe00:6f6b	NoUnit N741	MAC:76D74111012003B4	Voltage (mV)	2858
Tue Nov 20 10:11:23 CET 2012	/2001:660:5301:18:0:ff:fe00:6f6b	NoUnit N741	MAC:76D74111012003B4	Voltage (mV)	2878

Figure 45 L'onglet NanosensorTab modifié

1.3 Conclusion

La gestion des données des Nanosensors constitue la base de notre travail puisque d'une part c'est un des objectifs fixés au démarrage du stage et d'autre part, sans visibilité sur les données nous ne pourrions travailler ni sur l'aspect qualité de service, ni sur l'optimisation de la consommation d'énergie.

Dans cette section, nous avons présenté nos contributions dans la programmation des Nanosensors et les modifications apportées à l'application NodeView 2.0.

Dans la section qui va suivre, nous allons présenter nos travaux, expérimentations et conclusions autour de la qualité de service dans le réseau de capteurs Sensinode.

2 La qualité de service

Le deuxième objectif fixé au début de ce stage est l'étude et l'amélioration du support de la qualité de service. Cela consiste à faire des tests et des évaluations du matériel pour pouvoir ensuite proposer des solutions.

Nous avons pu déterminer suite à l'état de l'art que nous avons réalisé que la qualité de service peut être impactée par plusieurs facteurs qui peuvent être liés soit au matériel utilisé, soit aux protocoles implémentés mais aussi à l'environnement.

Dans le cas de notre réseau de capteurs Sensinode, nous avons décidé de commencer par faire des tests afin de déterminer les facteurs qui impactent le plus la qualité de service.

Dans cette partie nous allons présenter dans la première section les indices de qualité qui ont été définis pour évaluer le comportement du réseau par rapport aux facteurs de base qui peuvent impacter le niveau de service, la deuxième section comportera les différentes expériences, résultats, analyses et réactions. Nous consacrons la troisième section à l'aspect tolérance aux fautes et aux pannes et nous finirons par une conclusion.

2.1 Les indices de qualité

Pour déterminer les facteurs qui impactent le plus la qualité de service, nous avons défini plusieurs scénarii et réalisé différents tests. Pour chaque test, nous avons fixé des indices de qualité dans l'évaluation des résultats :

- La durée de vie (DDV) : il s'agit de la durée pendant laquelle le capteur est opérationnel, c'est-à-dire mesure, traite et envoie les données ;
- La couverture temporelle : c'est le taux des paquets reçus (PR) pendant la durée de vie. Par exemple, si le capteur envoie une donnée toutes les soixante secondes, nous devons recevoir 60 paquets par heure. On divise le nombre de paquets réels reçus par sa durée de vie, cela nous donne le taux moyen. Ensuite, on divise ce taux moyen par 60, ce qui nous donne le taux de couverture temporelle. La couverture temporelle est impactée à la fois par le taux de perte des paquets mais aussi en introduisant les opérateurs de sélection que nous avons définis ;
- Le taux de perte des paquets (TDP): il s'agit du ratio entre nombre de paquets reçus et le nombre de paquets attendus (PA). Par exemple, nous devons recevoir 60 paquets par heure, nous n'en recevons que 58, le taux de perte est égal à $(1 - (58/60)) = 0.004 \%$. D'une manière générale, le taux de perte est représenté par l'équation 1 :

$$TDP = 1 - (PR/PA)$$

Équation 1 : Taux de perte

- Le délai de transmission (DT): c'est le temps mis par un paquet pour arriver de la source à sa destination. Ceci peut traduire de cette façon (cf. équation 2) :

$$DT = TpsA - TpsD$$

Équation 2 : Délai de transmission

TpsA étant le temps d'arrivée et TpsD le temps de départ ;

- La gigue : il s'agit de la variation dans le délai de transmission. Nous pouvons déterminer la gigue minimum, maximum et moyenne. L'équation 3 permet de calculer la gigue moyenne :

$$GigueMoy = \frac{\sum_{i=1}^n (DT_{i-1} - DT_i)}{n-1}$$

Équation 3 : Gigue moyenne

Une fois les indices de qualité calculés et les résultats analysés, nous déterminons à quel point le facteur testé joue sur la qualité de service du réseau 6LoWPAN de Sensinode et proposons par la suite, si cela est possible, des techniques pour améliorer son comportement.

2.2 Détermination des facteurs impactant la qualité de service

Nous avons choisi d'évaluer le niveau de service par rapport à la distance entre les capteurs, les piles utilisées, la topologie du réseau et la couverture réseau.

2.2.1 Distance entre les Nanosensors et le Nanorouter

Ce test consiste à placer les Nanosensors à des distances différentes du Nanorouter.

Scénario

Nous avons placé deux capteurs à deux mètres du routeur et un troisième capteur à quinze mètres. Les trois capteurs envoient le niveau de batterie toutes les soixante secondes. Le résultat de cette expérience est donné dans la figure 46.

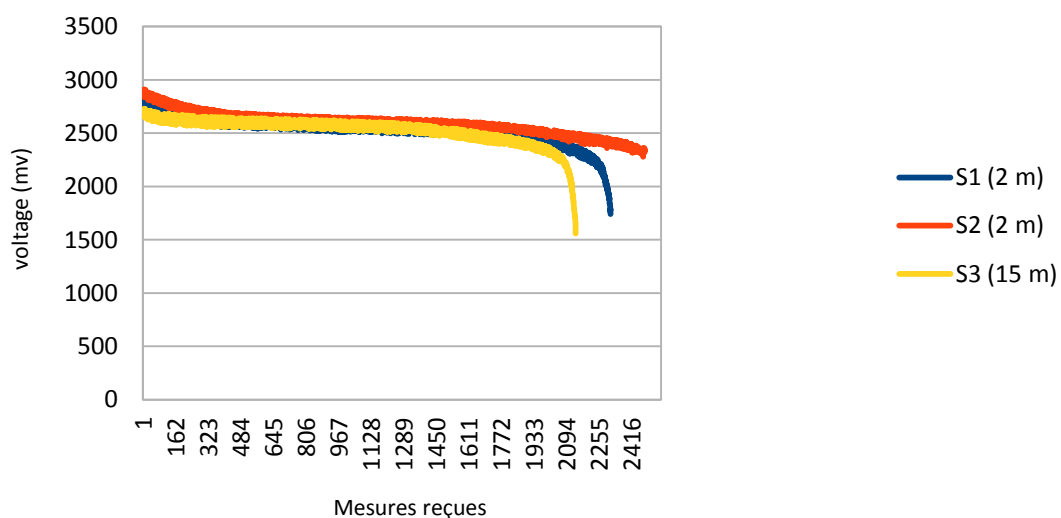


Figure 46 Impact de la distance sur la Qualité de service

Analyse des résultats

La durée de vie (cf. tableau 4) des deux premiers capteurs qui sont placés à deux mètres du routeur est supérieure à celle du troisième capteur.

En effet nous avons :

Tableau 4 Distance : indices de qualité

Capteur	Distance (m)	Durée de vie (h)	Taux de perte (%)	Couverture temporelle (%)
S1	2	39	0.5	99,5
S2	2	41	0.3	99.71
S3	15	35	1	99.07

Nous constatons aussi que le taux de perte des paquets est plus important pour le troisième capteur et que la couverture temporelle du réseau est plus satisfaisante pour les deux premiers capteurs même si dans l'ensemble les résultats sont très positifs. Le délai de transmission et la gigue ne sont pas présentés dans ce tableau car ces derniers étaient de l'ordre de microsecondes et pour des capteurs de luminosité ou de batterie, ceci reste acceptable.

D'après les résultats de ce test, nous avons pu déterminer que la distance à un impact très faible sur la qualité de service. Cela est dû principalement à la puissance du signal puisque plus la distance est grande plus la puissance du signal est importante et par conséquent les capteurs consomment plus d'énergie ce qui réduit la durée de vie.

2.2.2 Les piles utilisées

Scénario

Pour ce test, nous avons réalisé deux fois la même expérience avec les mêmes piles que nous avons rechargées entre temps. Les capteurs utilisés envoient comme mesures le niveau de batterie toutes les 120 secondes et sont placés à 2 mètres du Nanorouteur. Cette expérience s'est passée en deux temps et à chaque fois nous avons mesuré les indices de qualité définis précédemment, ensuite nous avons comparé les deux résultats. La figure 47 présente les mesures relevées pendant le test 1.

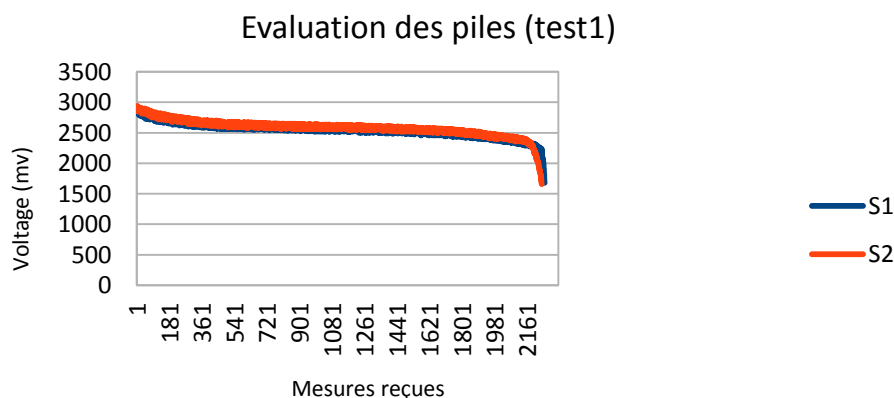


Figure 47 Impact des piles sur la qualité de service (test 1)

Nous constatons à ce niveau que la durée de vie des capteurs est presque égale. Au bout de ce test, nous avons rechargé les mêmes piles et relancé la même expérience, le comportement des capteurs est illustré dans la figure 48.

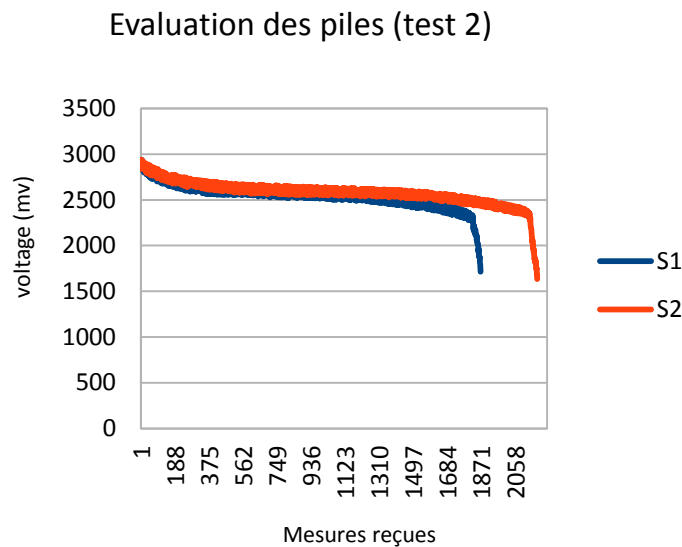


Figure 48 Impact des piles sur la qualité de service (test 2)

D'après la courbe du test 2, nous pouvons remarquer qu'il y a un écart au niveau de la durée de vie entre les deux capteurs. La première étant de 31h29 et la deuxième est égale à 36h40. Le délai de transmission et la gigue sont quasi-nuls. L'ensemble des indices de qualité calculés pour les deux tests est présenté dans le tableau 5.

Tableau 5 Piles et indices de qualité

	Capteur	Distance (m)	Durée de vie (h)	Taux de perte (%)	Couverture temporelle (%)
Test 2	S1	2	37h46	0.4	99,69
	S2	2	37h36	0.3	99.4
	S1	2	31h29	1.5	98.5
	S2	2	36h40	1	99.2

D'après ce tableau, même si le taux de perte des paquets ainsi que la couverture temporelle restent très satisfaisants, nous constatons un comportement très différent au niveau de la durée de vie. Pour le même capteur S1, il y a plus que six heures de différence ce qui est énorme et pour S2, il y a à peu près une heure.

Après l'analyse des résultats des deux tests, nous pouvons affirmer que les batteries utilisées impactent d'une manière conséquente la qualité de service.

2.2.3 La topologie du réseau

Afin d'évaluer la qualité de service par rapport à la topologie du réseau, nous avons placé les capteurs de telle sorte à avoir une architecture multisauts (cf. figure 49). Le but étant de comparer les indices de qualité liés à un capteur jouant le rôle d'une gateway et celles des

capteurs placés à deux sauts du Nanorouteur et passants par le « capteur-gateway ». Ensuite nous avons comparé les résultats de ce scénario à ceux d'une architecture plate où tous les capteurs sont à 1 saut.

Scénario : nous avons utilisé trois capteurs, le premier joue le rôle de capteur-gateway et les deux autres sont placés derrière. L'ensemble envoie le niveau de batterie toutes les 60 secondes.

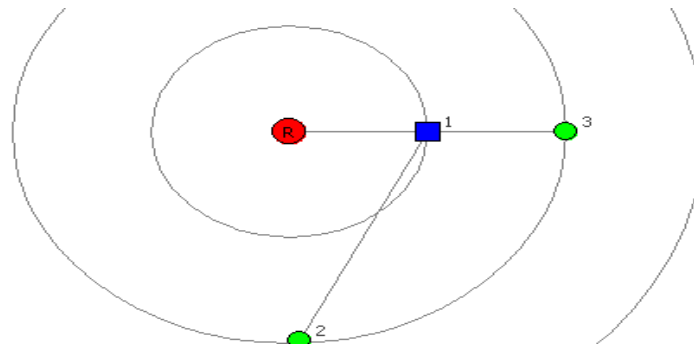


Figure 49 Architecture Multisauts 3 capteurs

A l'issue de cette expérience, nous avons tracé cette courbe (figure 50) qui présente le nombre de mesures envoyées par chaque capteur.

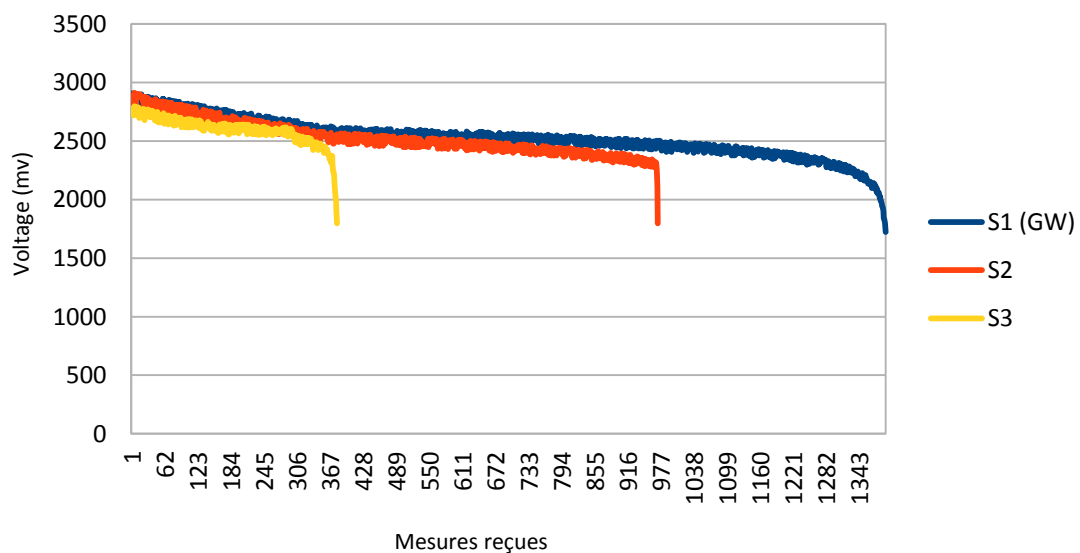


Figure 50 Impact de la topologie du réseau sur la qualité de service (1)

Nous constatons d'après ce graphe que le taux de perte des paquets pour le capteur 3 est très important par rapport à celle des deux autres. En effet ce dernier était le plus éloigné.

L'ensemble des indices de qualité est présenté dans le tableau 6.

Tableau 6 Topologie et indices de qualité (1)

Capteur	Distance (m)	Durée de vie (h)	Taux de perte (%)	Couverture temporelle (%)
S1 (Gw)	20	38h16	39.4	60.6
S2	25	32h31	50.2	49.8
S3	35	21h41	70.8	29.31

D'après le tableau ci-dessus, nous constatons que les meilleurs indices de qualité sont retournés par le capteur-gateway mais nous avons quand même des pourcentages très élevés et la qualité de service est très dégradée. La durée de vie du capteur S3 ne dépasse pas les 22 heures alors que d'après les expériences précédentes, nous savons que les piles peuvent tenir jusqu'à 42 heures. La durée de vie des deux autres capteurs restent dans la moyenne. Au niveau du taux de perte de paquets, nous avons des taux importants et la couverture temporelle est très basse.

Après l'analyse de ces chiffres, nous expliquons ce comportement par :

- Le capteur-gateway présente une situation plus stable que les autres parce qu'il n'est qu'à un saut du routeur ;
- La distance joue sur la durée de vie et la puissance du signal c'est ce qui a conduit à des durées de vie assez limitées.
- Les deux autres capteurs, vu qu'ils sont à deux sauts, cherchent continuellement à garder le lien avec le capteur-gateway et en même temps de joindre directement le Nanorouteur. Nous montrons dans le tableau 7 l'instabilité qu'a subi le réseau dans une telle architecture ;

Tableau 7 Changement d'adresses IP

Capteur	Changement d'adresse IP
S1 (Gw)	4 fois
S2	9 fois
S3	14 fois

- La qualité de service est mieux supportée dans une architecture plate que dans une architecture multisauts.

Au terme de cette expérience, nous pouvons affirmer que la topologie du réseau impacte considérablement le niveau de service de notre réseau de Nanosensors.

Améliorations : Rappelons que cette expérience a été réalisée avec trois capteurs seulement alors nous pensons que les indices de qualités peuvent être améliorés si nous modifions le paramètre densité du réseau. Il s'agit de déployer plusieurs capteurs dans une même zone. Ceci permettra de répartir la charge sur les gateway et offrir plus de chemins pour acheminer les paquets.

Nous avons réalisé une deuxième expérience selon le scénario suivant. Nous utilisons dix Nanosensors au lieu de trois. Les distances entre les capteurs ne dépassent pas les dix mètres. La figure 51 montre l'architecture du réseau suite à ce changement.

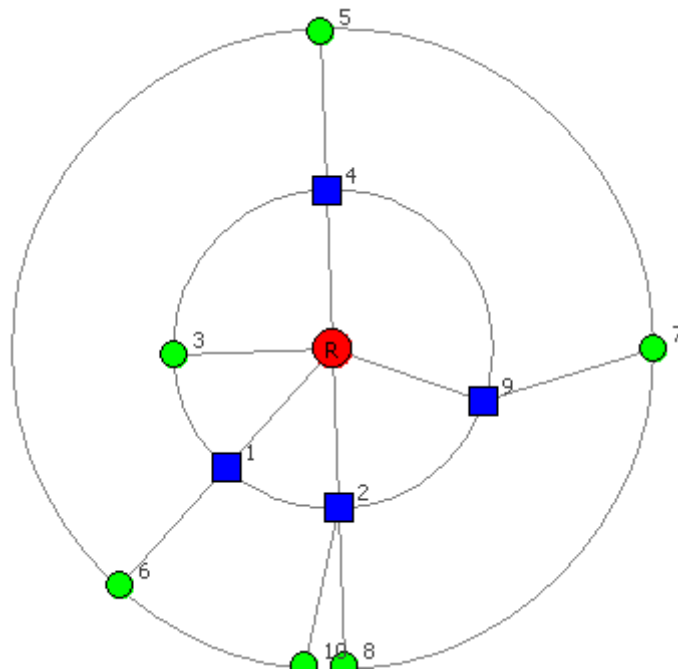


Figure 51 Architecture multisauts 10 capteurs

Nous constatons que grâce à la technologie 6LoWPAN, quatre Nanosensors jouent le rôle de gateway, ceci est dans le but de partager la charge du réseau. Nous avons constaté des changements de la topologie plusieurs fois tout au long de cette expérience.

Nous avons tracé la courbe qui représente le nombre de mesures envoyées par chaque capteur (cf. figure 52), nous constatons une grande différence par rapport à celle du premier scénario.

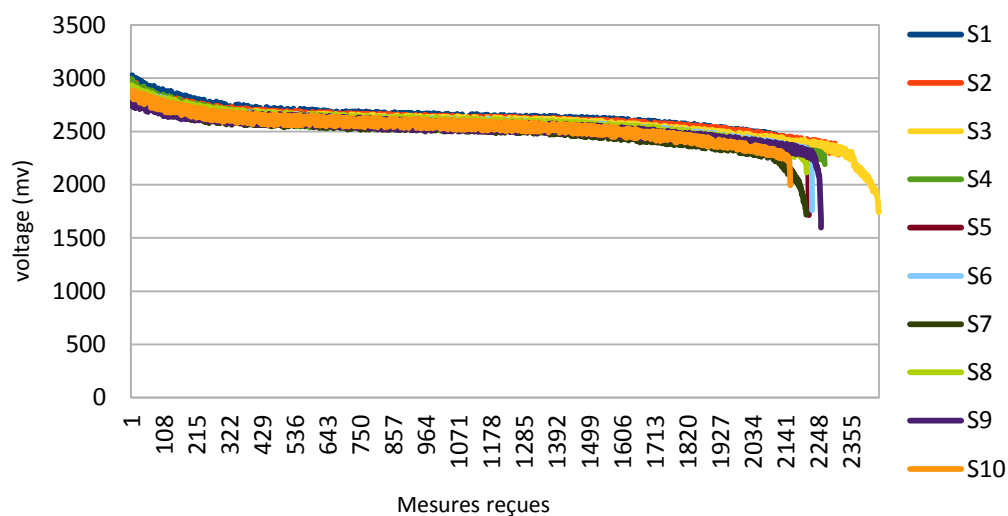


Figure 52 Impact de la topologie du réseau sur la qualité de service (2)

Les indices de qualité sont illustrés dans le tableau 7.

Tableau 8 Topologie et indices de qualité (2)

Capteur	Durée de vie (h)	Taux de perte (%)	Couverture temporelle (%)
S1	35h03	1,12%	98,88%
S2	39h14	3,26%	96,74%
S3	39h51	0,97%	99,03%
S4	37h12	1,30%	98,70%
S5	37h34	2,20%	97,80%
S6	36h36	3,55%	96,45%
S7	37h41	3,49%	96,51%
S8	36h31	2,04%	97,96%
S9	37h52	1,78%	98,22%
S10	35h45	2,93%	97,07%

D'après la courbe 52 et le tableau 8, nous pouvons affirmer que la densité du réseau joue un rôle très important dans la qualité de service. En effet dans une architecture multisauts où on a une forte densité du réseau, le comportement de ce dernier devient presque similaire à une architecture plate. Le taux de perte de paquets ne dépasse pas 3,55% et la couverture temporelle dépasse les 96% pour tous les Nanosensors. Nous n'avons pas présenté les indices délai de transmission et gigue dans le tableau car ces derniers étaient de l'ordre de quelques millièmes de secondes. Ces valeurs sont très satisfaisantes dans notre contexte.

Après analyse des indices de qualité, nous pouvons affirmer que la distance entre les Nanosensors, les piles, la topologie du réseau et la densité de ce dernier impactent plus ou moins la qualité de service. C'est dans une architecture multisauts à faible densité que le réseau présente de grandes insatisfactions, la solution que nous avons proposé consiste à diminuer la distance entre les capteurs et augmenter la densité du réseau.

La tolérance aux fautes et aux pannes est aussi une des métriques principales à prendre en compte. Pour pouvoir évaluer la capacité de kit 6LoWPAN dans ce contexte, nous avons réalisé des expérimentations en se basant sur la mesure de la luminosité. Les scénarii, résultats et analyses sont détaillés dans la section suivante.

2.3 Tolérance aux fautes et qualité de service

Il existe plusieurs types de fautes dans les réseaux de capteurs. Le matériel mis à notre disposition nous permet d'en évaluer deux. Il s'agit des fautes de valeurs et des pannes matérielles. Concernant les valeurs, nous avons défini des tests dont le but est de pouvoir comparer les valeurs en provenance de différents capteurs couvrant la même zone. Pour les pannes matérielles, nous avons étudié l'aspect couverture réseau.

2.3.1 Tolérance aux fautes de mesures

1^{er} cas : deux capteurs dans une architecture en étoile

Dans ce cas, nous avons placé deux capteurs côte à côte à deux mètres du Nanorouter, ils envoient les mesures de luminosité toutes les deux secondes. Le but est de comparer les

valeurs reçues et en déduire s'il y a présence ou non des fautes de mesures. La figure 53 présente les données reçues.

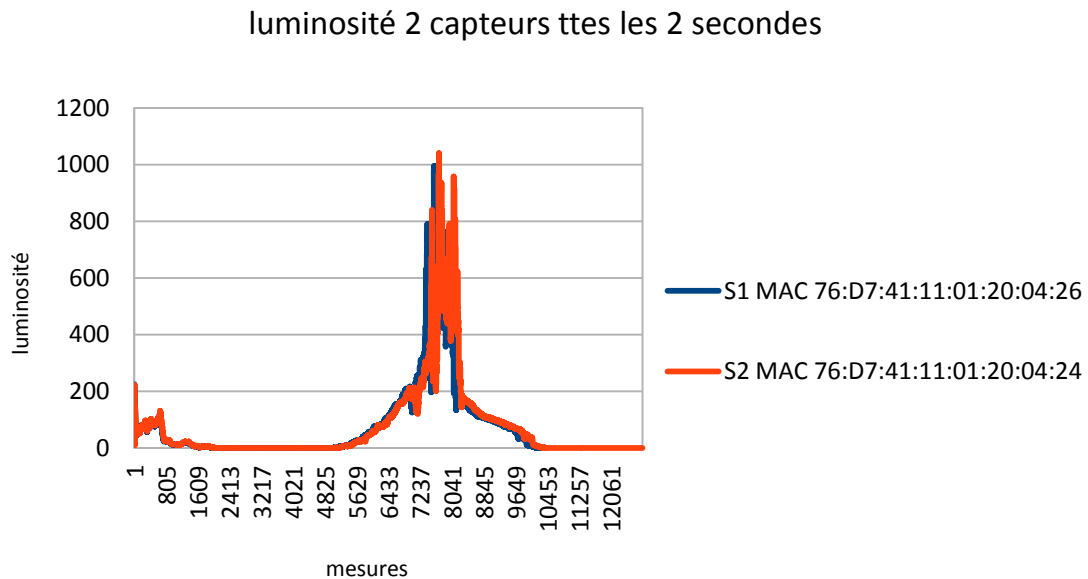


Figure 53 Mesures de luminosité dans une architecture en étoile utilisant deux capteurs

Nous remarquons une quasi-superposition des deux courbes, les deux capteurs envoient des valeurs presque égales, le nombre de paquets reçus qui est en rapport avec la durée de vie est presque égale. Nous en déduisons que dans une telle configuration, les fautes présentes peuvent être tolérées parce que d'une part les valeurs se rapprochent et d'autre part la mesure de la luminosité est très sensible. Nous avons calculé, pour une même date, la variation (V_{mesure}) des mesures (cf. figure 54). Celle-ci ne dépasse pas 3 mV ce qui est négligeable. Nous avons :

$V_{\text{mesure_min}} = 0 \text{ mV}$; $V_{\text{mesure_max}} = 3 \text{ mV}$; $V_{\text{mesure_moy}} = 1,5 \text{ mV}$

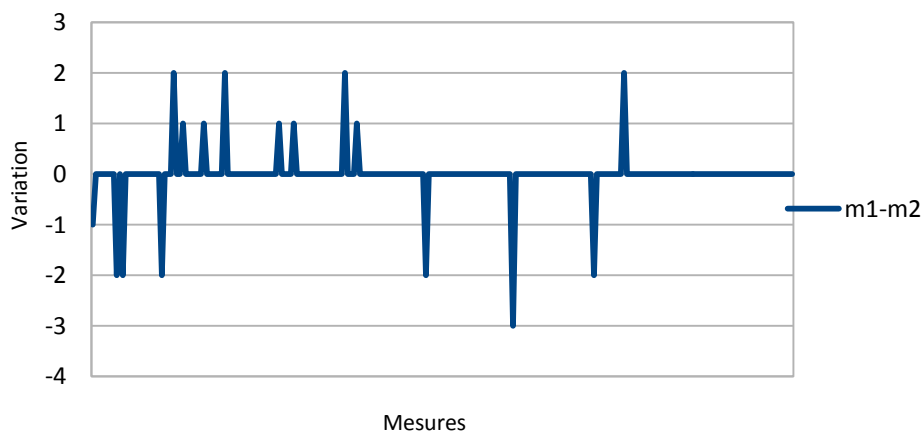


Figure 54 Variation des mesures

2^e cas : trois capteurs dans une architecture étoile

Nous avons placé trois capteurs côte à côte dans une pièce, ils envoient les mesures de luminosité toutes les trente secondes, le résultat est présenté dans la figure 55.

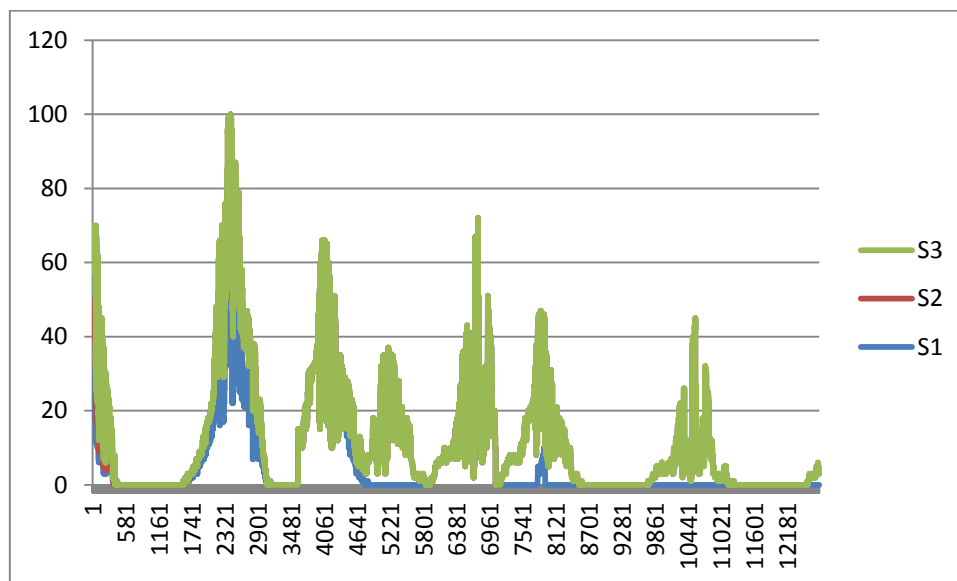


Figure 55 Mesures de luminosité dans une architecture en étoile utilisant trois capteurs

Nous avons presque le même comportement que le cas précédent. En effet, nous avons remarqué une variation plus importante mais nous considérons que les fautes sont tolérables. Par contre, au niveau de la durée de vie le capteur S2 s'éteint bien avant les deux autres.

Synthèse : d'après les résultats des deux expériences réalisées, nous pouvons affirmer que les Nanosensors ne souffrent pas de problèmes de fautes de valeurs. Même si s'était le cas, il y avait la possibilité de reprogrammer les microcontrôleurs MSP430F5438 pour faire un calibrage des mesures.

2.3.2 Tolérance aux pannes matérielles

Nous avons vu suite aux différentes expérimentations effectuées précédemment que parfois la durée de vie des capteurs n'est pas égale, ceci peut être dû aux piles, à l'architecture et à la distance qui sépare les Nanosensors du Nanorouteur. Nous définissons que le réseau rencontre une panne matérielle lorsque nous ne recevons plus de données de la part d'un capteur. Pour assurer un niveau de service satisfaisant, nous avons mis en place deux techniques afin de faire face aux pannes matérielles.

La redondance matérielle

Cette technique est simple à mettre en place mais a un coût financier puisque le fait de doubler ou tripler le nombre de capteurs demande de l'argent. Le but de cette technique est que si un Nanosensor est en panne de batterie ou n'arrive plus à se connecter au réseau, nous continuons à recevoir les données grâce aux autres capteurs de la même zone. La figure 55 montre que le capteur S2 n'envoyait plus de données depuis un certain temps mais les deux autres capteurs ont continué à le faire et la qualité de service a été assurée.

La couverture réseau

Pour faire face aux pannes matérielles, nous avons diminué les distances entre les capteurs pour que si un capteur tombe en panne, tous les autres peuvent continuer à communiquer entre eux et assurer la continuité du service. En effet, nous avons réalisé deux expériences où nous avons varié la distance entre les capteurs. Dans la première, la distance maximale est de dix mètres, dans la deuxième la distance varie entre vingt et trente mètres et à chaque fois nous avons éteint des capteurs intermédiaires pour voir la réaction des Nanosensors.

- **1^{er} cas : la distance entre les nœuds ne dépasse pas 10 mètres**

Lors de la première expérience, nous n'avons pas remarqué la panne. En effet nous avons continué à recevoir les données régulièrement. Le Nanosensor a détecté que la Gateway qu'il utilise n'est plus joignable par conséquent, grâce au mécanisme d'autoconfiguration de la technologie 6LoWPAN et au protocole de routage RPL, le capteur a découvert une autre gateway et a continué à émettre ses paquets.

Cette situation est présentée dans la figure 56.

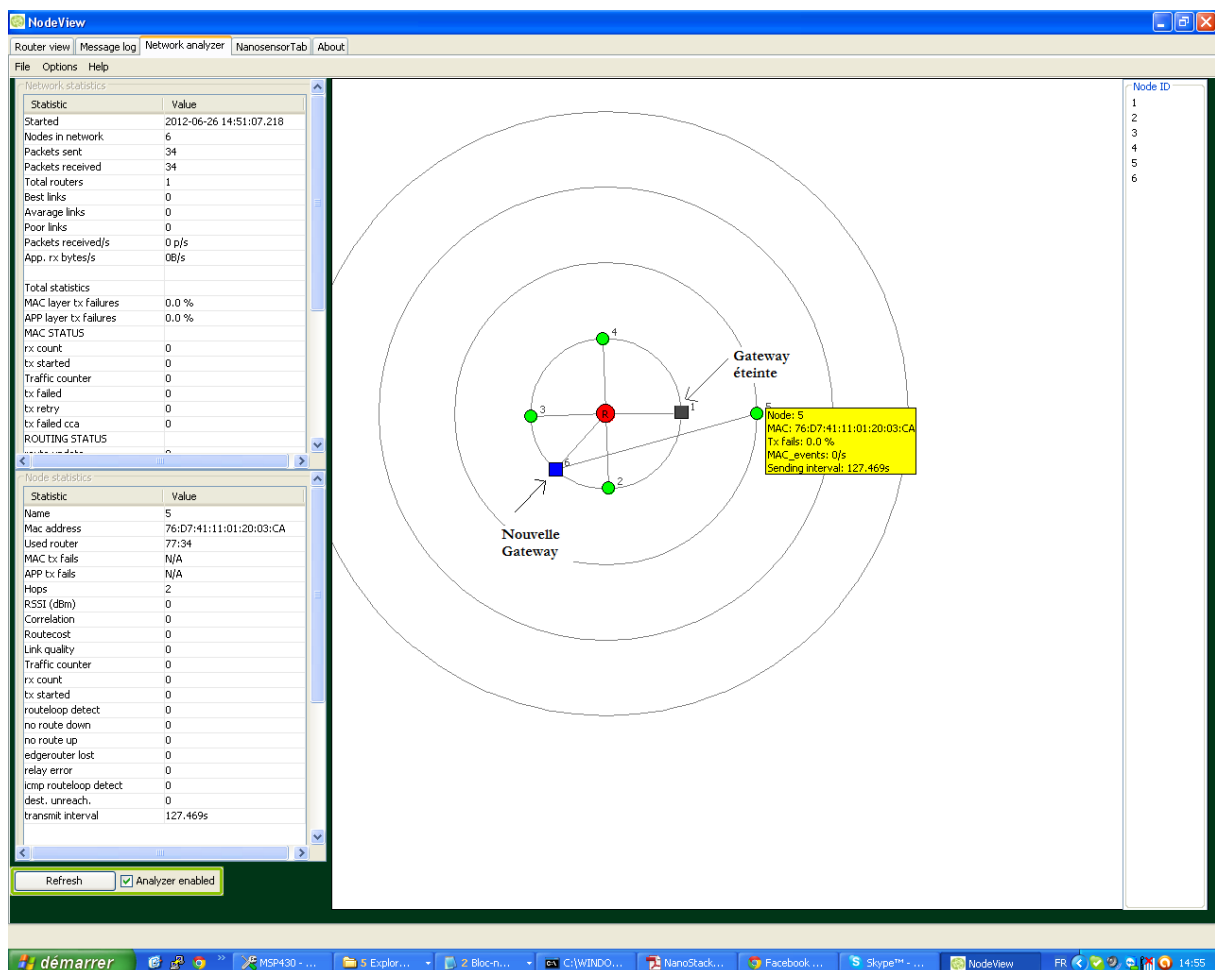


Figure 56 Tolérance aux pannes et couverture réseau (1)

- **2^e cas : la distance entre les nœuds est supérieure à 20 mètres**

Pendant le deuxième test, nous avons remarqué une dégradation du service. En effet, nous avons éteint un capteur qui jouait le rôle de gateway par conséquent le capteur se trouvant derrière n'arrivait plus à joindre les Nanorouteur et nous ne recevions plus de données. Ce scénario est illustré dans la figure 57.

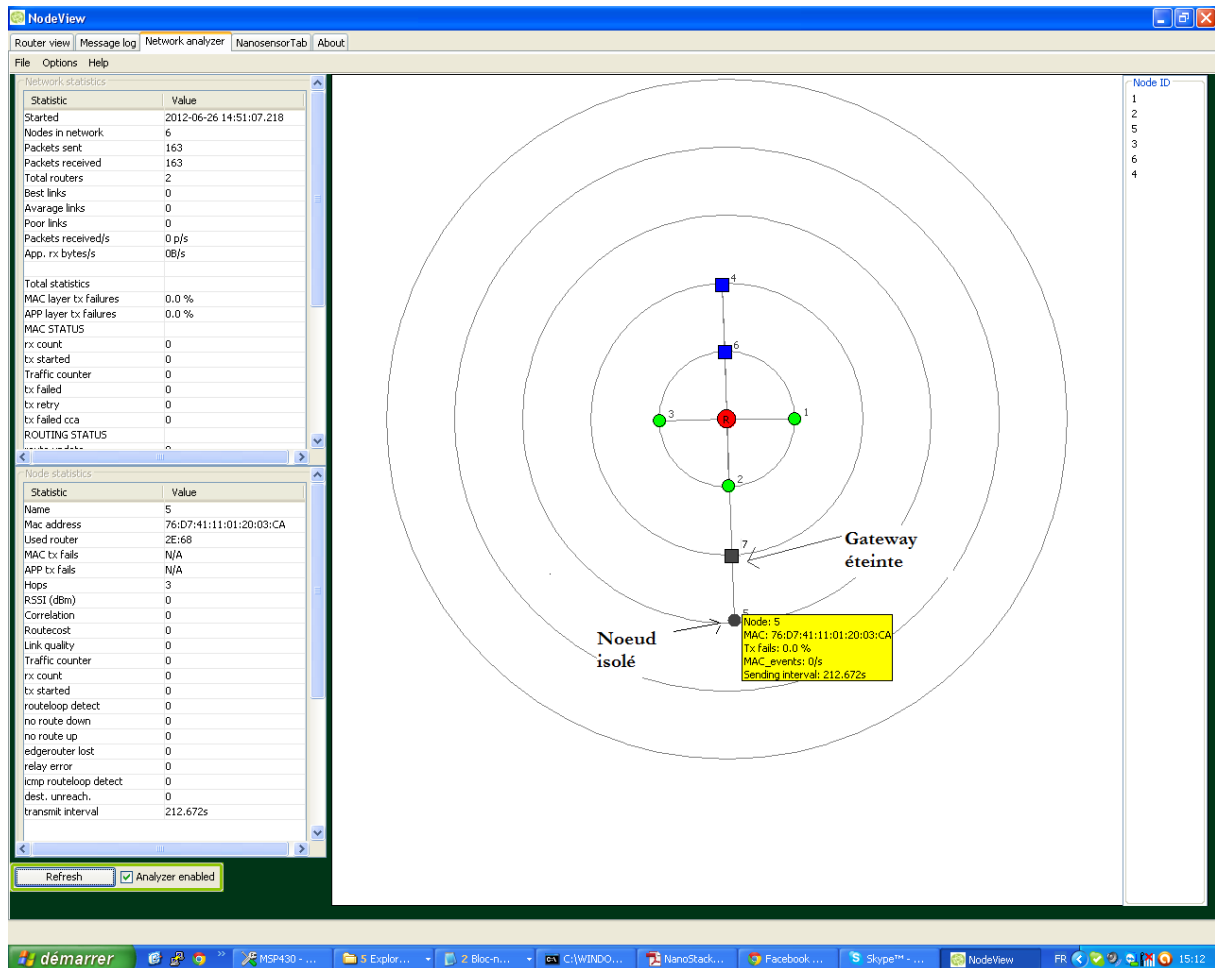


Figure 57 Tolérance aux pannes et couverture réseau (2)

Cette figure nous montre que le nœud 5 est devenu complètement isolé par conséquent il n'y a plus de couverture réseau dans cette zone. Dans cette situation, nous avons une dégradation très importante de la qualité de service.

Les pannes dans les réseaux de capteurs impactent plus ou moins la qualité de service. En effet cela dépend de la topologie, de la couverture réseau et du matériel utilisé.

2.4 Conclusion

Dans cette section, nous avons présenté notre étude, l'évaluation et les techniques qui ont été mises en place pour étudier la qualité de service dans le réseau de Nanosensors de Sensinode. Cette dernière est impactée par plusieurs facteurs qui sont la distance entre les capteurs, les pannes matérielles, l'architecture du réseau et les batteries utilisées. Nous avons remarqué tout au long des différentes expériences réalisées que la durée de vie des capteurs tourne autour de trente heures. La durée de vie est liée à la consommation

d'énergie des Nanosensors. Dans la section suivante, nous allons présenter les techniques que nous avons utilisées pour minimiser cette consommation d'énergie et prolonger la durée de vie du réseau.

3 Conservation d'énergie

Comme dans tous les réseaux de capteurs, la consommation d'énergie est un des facteurs les plus importants qui impacte la qualité de service. Cette ressource signifie la durée de vie du réseau et en général les applications à base de capteurs demandent qu'elle soit la plus longue possible. L'énergie est consommée plus ou moins par les différents modules des capteurs. L'unité d'acquisition et de traitement en consomme une partie conséquente mais la plus grosse quantité d'énergie est utilisée par l'unité de transmission.

Nous allons présenter dans la sous-section 1 nos travaux pour la conservation d'énergie au niveau du microcontrôleur (unité d'acquisition et de traitement), ensuite nous présenterons nos travaux et nos propositions au niveau de l'unité de transmission et nous finirons par une conclusion.

3.1 Conservation d'énergie au niveau de l'unité d'acquisition et de traitement

L'énergie consommée par le module d'acquisition et de traitement des données n'est pas celle qui limite le plus la durée de vie des Nanosensors. Néanmoins, nous avons évalué cette consommation et essayé de la réduire sans pour autant impacter la qualité des données et la qualité de service.

3.1.1 Eteindre les capteurs non utilisés

Nous voulons dire par cela que nous n'avons gardé opérationnels que les capteurs qui nous ont servi à la fois de pouvoir utiliser l'application NodeView 2.0 et mesurer la qualité de service. Il s'agit des capteurs de luminosité et de niveau de batterie. Nous avons fait ce choix et éteint le capteur d'accélération lors de toutes nos expériences. L'opération consiste en la désactivation des canaux 0 à 2 du convertisseur analogique numérique de l'ADC12, la non création des messages relatifs à l'accélération et l'arrêt d'envoi des données. Cette démarche ne nous a pas permis de conserver une quantité d'énergie conséquente mais la durée de vie a été rallongée de quelques minutes.

3.1.2 Minimisation de la fréquence d'échantillonnage

Dans ce cadre, nous avons introduit une nouvelle fonction au niveau du code MSP430 afin de mettre au repos le microcontrôleur pendant la période où nous n'avons pas besoin de données. Il s'agit de la fonction `__delay_cycles(xx)`, `xx` étant la durée pendant laquelle l'ADC12 n'effectue aucune opération c'est-à-dire arrêter l'échantillonnage et la conversion des données. Nous nous sommes attendus à ce que le résultat soit conséquent mais ceci n'était pas le cas. La durée de vie continue à avoisiner les trente heures. Nous expliquons ceci par le fait que même si les capteurs envoient moins de messages, la conversion au niveau de l'ADC est maintenue.

3.1.3 Utilisation des modes basse consommation

Nous avons présenté dans le chapitre 2 les principales caractéristiques du microcontrôleur MSP430F5438. Celle que nous estimons efficace dans la conservation d'énergie des Nanosensors est la possibilité d'utiliser cinq modes de basse consommation (cf. page 46). Il s'agit à chaque fois d'activer ou désactiver des circuits, horloges ou autres. Nous avons introduit ces fonctions dans notre code et réalisé des expériences pour pouvoir mesurer l'évolution de la durée de vie.

Le principe est le même pour les cinq modes, nous utilisons une fonction qui permet d'entrer dans le mode basse consommation après la fin de conversion des données et nous introduisons une deuxième fonction qui permet de sortir de ce mode (cf. figure 58) une fois que la donnée est stockée dans la mémoire. Les deux fonctions introduites dans le fichier *adc_driver.c* sont :

- `__bis_SR_register(LPMx_bits + GIE)` : pour entrer dans le mode LPMx;
- `__bic_SR_register_on_exit(LPMx_bits)` : pour sortir du mode LPMx.

x étant le numéro du mode, il varie de 0 à 4.

```
ADC12CTL0 |= ADC12ENC; // Enable conversions

ADC12CTL0 |= ADC12SC; // Start conversion moh
__bis_SR_register(LPM0_bits + GIE); // Enter LPM0
...

#pragma vector=ADC12_VECTOR
__interrupt void ADC12ISR (void)
{
    switch(__even_in_range(ADC12IV,34))
    {
        #if 0
        case 0:
            break; // Vector 0: No interrupt
        case 2:
            break; // Vector 2: ADC overflow
        case 4:
            break; // Vector 4: ADC timing overflow
        #endif
        case 6:
            AD_value = ADC12MEM0;
            AD_value = 5 * AD_value;
            AD_value = AD_value >> 2; // should be mV
            AD_ready = 1;
            __bic_SR_register_on_exit(LPM0_bits); // Exit active CPU
    }
}
```

Figure 58 Entrer et sortir du mode LPMx

Mode LPM0

Afin d'évaluer ce mode, nous avons utilisé un Nanosensor qui envoie le niveau de batterie toutes les 60 secondes, il est placé à deux mètres du Nanorouter.

L'expérience a duré 42 heures, c'est la première fois que nous atteignons cette durée de vie, nous sommes arrivés à gagner 12 heures par rapport à la moyenne qui est 30 heures. Par ailleurs, nous remarquons que l'allure de la courbe est restée la même que toutes les expériences précédentes c'est-à-dire la chute est brutale vers la fin de vie de la batterie.

Mode LPM1

Nous avons utilisé le même scénario que l'expérience précédente c'est-à-dire envoyer le niveau de batterie, mais nous avons changé la fréquence, en effet les mesures sont envoyées toutes les dix minutes. L'expérience en mode LPM1 a duré 39 heures et trente minutes, le gain cette fois-ci est de 9 heures et trente minutes par rapport à la moyenne. L'allure de la courbe est la même que précédemment et la chute est brutale vers la fin.

Mode LPM2

La durée de vie est égale à 39h20. Comme en LPM1, le gain est de 9 heures et vingt minutes et l'allure de la courbe reste la même (cf. figure 59).

Mode LPM3

Toujours le même scénario que les autres expériences et le résultat est donné dans la figure 59. La durée de vie est égale à celle atteinte en LPM1 c'est-à-dire 39h30.

Mode LPM4

Tout comme en LPM1 et LPM3, nous avons une durée de vie égale à 39h30 et l'allure de la courbe est la même.

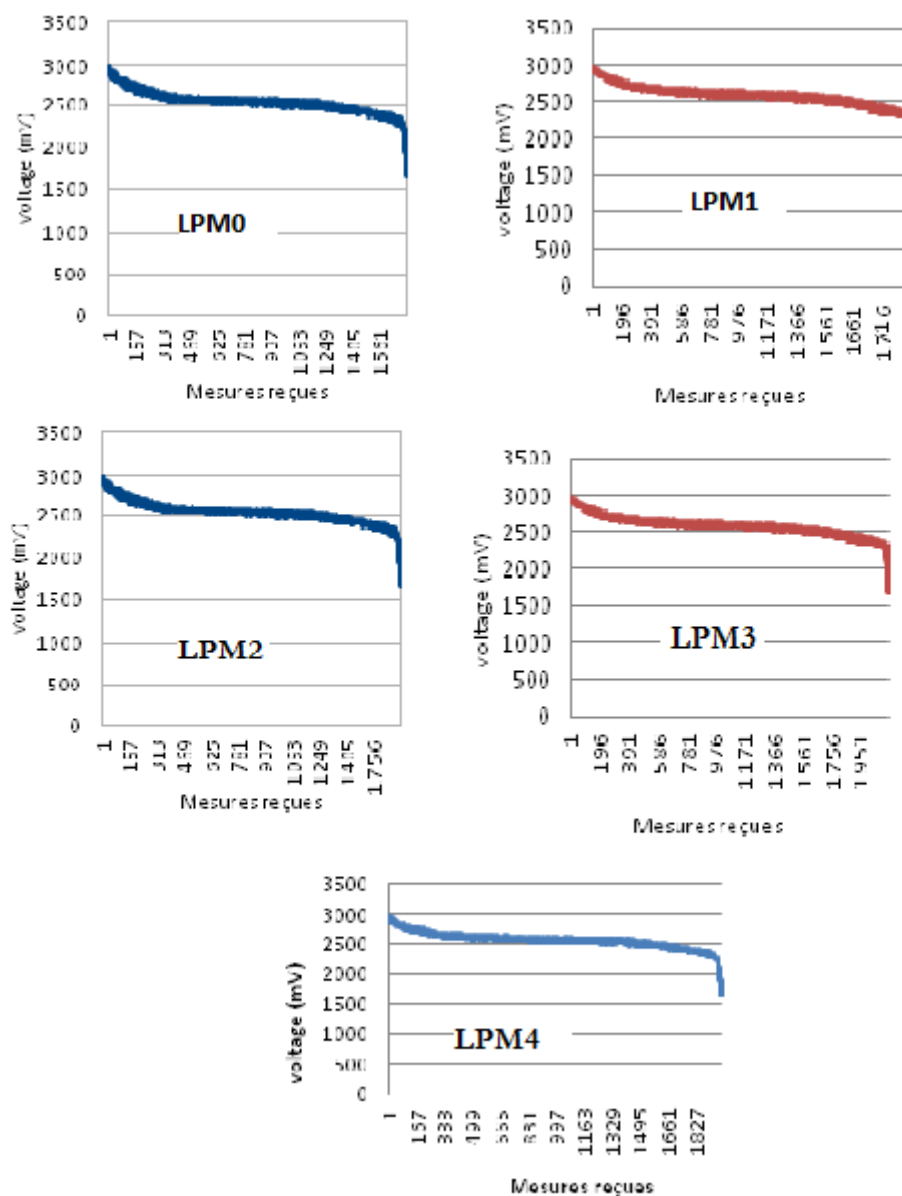


Figure 59 Energie en mode LPM

A part le mode LPM0 avec lequel nous sommes arrivés à une durée de vie égale à 42 heures, tous les autres modes nous ont permis d'atteindre 39 heures pourtant en mode LPM0, il y a moins de composants désactivés et logiquement il doit être le mode avec lequel nous arrivons à moins gagner en durée de vie. Nous expliquons cela par le fait qu'en utilisant les modes LPM 1 à 4, certes il y a plus de composants qui sont désactivés par rapport à LPM0 mais le fait de les « réveiller » demande aussi plus d'énergie.

Les résultats des techniques et les expériences que nous avons réalisées afin de conserver l'énergie au niveau de l'unité de acquisition et de traitement de données nous ont confirmé que ce n'est pas cette dernière qui consomme la plus grande quantité vu que le gain maximum que nous avons pu atteindre ne dépasse pas 25% de la durée de vie des Nanosensors. Nous allons présenter dans la section suivante nos contributions relatives à la conservation d'énergie au niveau de l'unité de transmission.

3.2 Conservation de l'énergie au niveau de l'unité de transmission

L'unité de transmission des Nanosensors est composée du module radio RC2400HP de Radiocrafts plus une antenne multidirectionnelle. Nos contributions touchent les deux parties.

3.2.1 Réduction de la puissance d'émission

Après plusieurs recherches, nous avons trouvé sur le site du constructeur du module radio www.radiocrafts.com un outil qui permet de configurer certaines fonctionnalités de ce dernier. Celle qui nous intéresse est la gestion de la puissance d'émission. L'outil en question s'appelle ZNM-CCT.

3.2.1.1 ZNM-CCT

ZNM-CCT est un outil qui a été développé pour permettre de configurer les modules radio dans les réseaux de capteurs utilisant la technologie Zigbee et il est compatible avec notre matériel Sensinode. Cet outil permet de se connecter au capteur, de lire les données en provenance de ce dernier et de programmer le composant Radiocrafts en utilisant des commandes stockées dans des fichiers de configuration. Un aperçu de ce fichier est donné dans la figure 60.

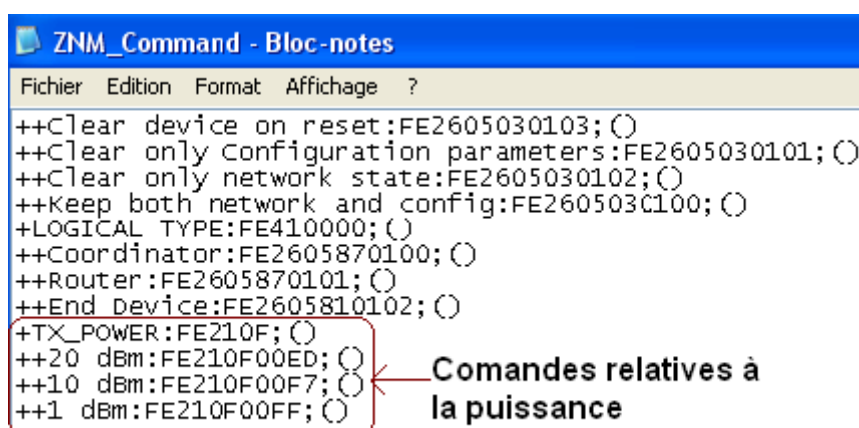


Figure 60 Fichier de commande ZNM-CCT

La consommation d'énergie relative à chaque puissance a été donnée dans le tableau 3 au niveau du chapitre 2. Nous avons choisi trois niveaux de puissances, 1, 10 et 20 dBm pour faire les tests. Le paramètre à varier est TX_POWER.

Dans la figure 61, nous présentons les différentes étapes pour modifier ce paramètre.

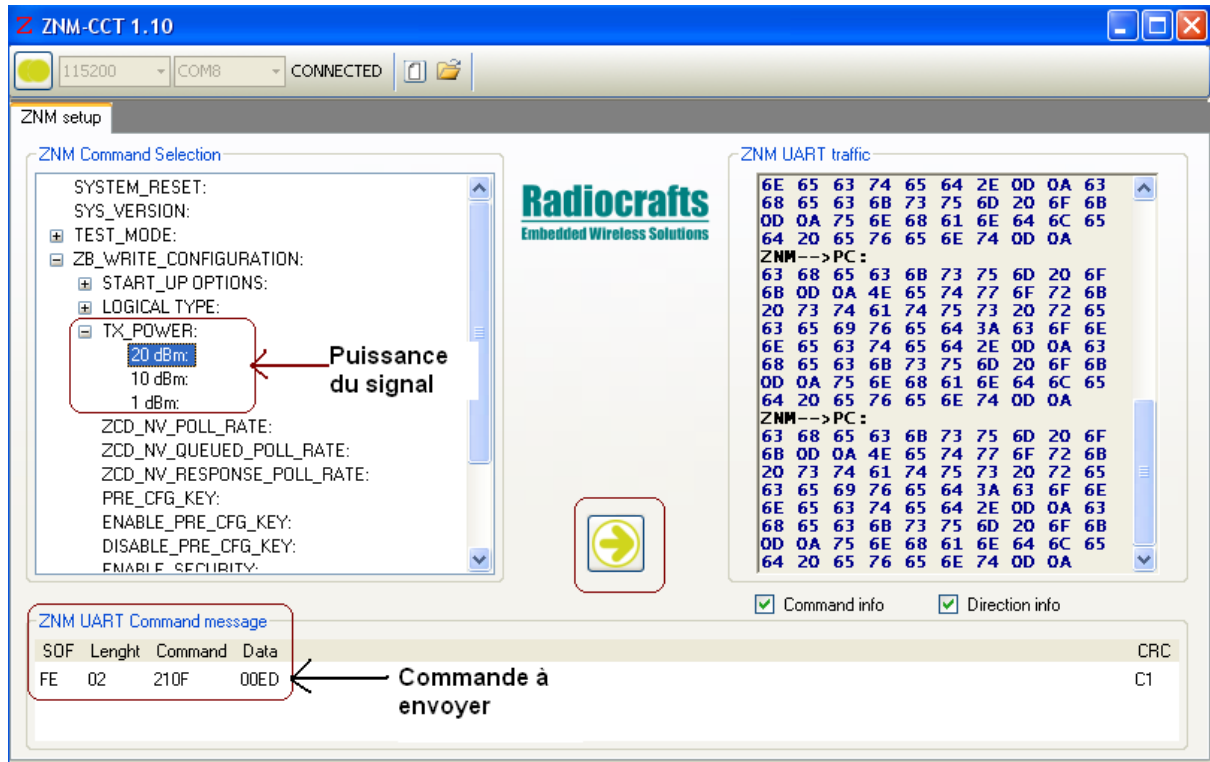


Figure 61 ZNM-CCT

3.2.1.2 Expériences et résultats

Logiquement, plus la puissance est faible moins on consomme d'énergie et la durée de vie est augmentée. Nous avons placé deux Nanosensors, le premier à deux mètres du Nanorouter et le deuxième à dix mètres. Le but d'introduire la distance dans l'expérience est de confirmer son impact sur le taux de perte des paquets et la qualité de service. Les deux capteurs envoient le niveau de batterie toutes les minutes.

En utilisant une puissance égale à 20 dBm, nous constatons que S1 situé à 2 mètres a envoyé plus de mesures que S2 situé à 10 mètres, nous en déduisons qu'il y a un taux de perte de paquets plus important chez S2. Concernant la durée de vie, celle de S1 est égale à 37 heures et celles de S2 est égale à 36. Nous avons un gain de 6 heures par rapport à la moyenne qui est de 30 heures (cf. tableau 9).

Pour une puissance du signal égale à 10 dBm, nous avons constaté un taux de perte de paquets moindre chez S1 par rapport à S2. Concernant la durée de vie, celle de S2 s'élève à 38 heures et celle de S1 atteint 40 heures (cf. tableau 9). Nous arrivons à conserver un peu plus d'énergie par rapport à un TX_POWER égale à 20 dBm.

En réduisant la puissance du signal à 1 dBm, nous sommes arrivés à atteindre une durée de vie égale à 43 heures au niveau de S1 et 41 heures au niveau de S2 (cf. tableau 9). S1 a également assuré un taux de délivrance de paquets plus important que S2.

Nous présentons l'impact de la puissance du signal sur la durée de vie et le taux de perte des paquets dans le tableau 9.

Tableau 9 Puissance du signal vs durée de vie et taux de perte

Puissance (dBm)	Durée de vie (h)		Taux de perte (%)	
	S1 (à 2 m)	S2 (à 10 m)	S1 (à 2 m)	S2 (à 10 m)
20	37	36	1	1.5
10	40	38	0.9	1.6
1	43	41	1	1.9

La figure 62 récapitule les résultats des différentes expériences.

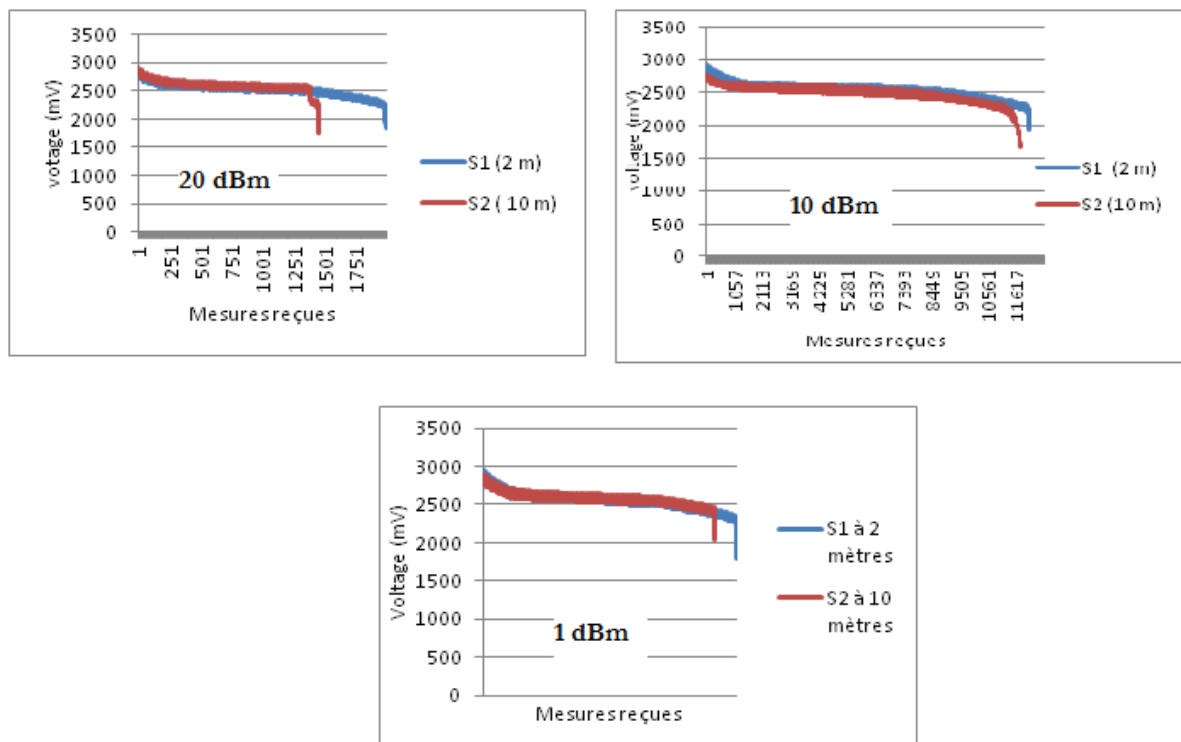


Figure 62 Varier la puissance du signal

D'après les résultats des trois expériences réalisées, nous avons constaté que le fait de varier la puissance du signal agit sur la consommation d'énergie. En effet plus la puissance est faible, plus longue est la durée de vie. Par contre le taux de perte des paquets est aussi impacté c'est-à-dire, en utilisant un TX_POWER très faible, les paquets de données ont du mal à atteindre la destination. Même si nous avons réussi à gagner quelques heures, ceci reste insuffisant, nous allons passer aux travaux réalisés au niveau du module radio pour essayer de gagner encore plus en durée de vie.

3.2.2 Conservation d'énergie au niveau du module radio

Nous rappelons que le module radio utilisé, RC2400HP de Radiocrafts, est composé d'un SoC CC2530 plus un CPU 8051 de Texas Instrument. Afin de réduire la consommation

énergétique au niveau du module radio, nous avons choisi de mettre en œuvre la technique Sleep/Wakeup déjà présentée dans le chapitre état de l'art de ce mémoire. Pour arriver à utiliser cette technique, il faut agir sur le SoC CC2530 et programmer les différentes périodes de sommeil et de réveil. Le matériel mis à notre disposition ne nous a pas permis d'utiliser cette technique en pratique parce que le SoC est verrouillé et le flasher s'avère impossible. Pour cette raison, nous avons choisi de simuler cette démarche vu que nous avons toutes les caractéristiques techniques du RC2400HP en particulier les quantités d'énergie consommées relatives à chaque état dans lequel se trouve le capteur c'est-à-dire émission, réception, idle ou endormi. Nous rappelons ces valeurs ci-dessous :

- Active-Mode **RX** (CPU Idle): **24 mA**
- Active Mode **TX** (CPU Idle): **29 mA**
- **Power Mode 1** (4 μ s Wake-Up): **0.2 mA**
- **Power Mode 2** (Sleep Timer Running): **1 μ A**
- **Power Mode 3** (External Interrupts): **0.4 μ A**

Nous constatons qu'il n'y a pas une grande différence de consommation énergétique entre le mode émission (TX) et le mode réception (RX) mais afin de se rapprocher le plus possible des valeurs réelles, nous avons calculé la moyenne entre ces deux modes. Pour cela, nous sommes basés sur la fréquence d'échantillonnage que nous avons utilisée le plus dans la plupart de nos expériences précédentes, c'est-à-dire 60 secondes et une durée de vie moyenne (DDVM) égale à 30 heures. Nous avons mis en place plusieurs scénarii en tenant compte à chaque fois de plusieurs paramètres et pour simplifier le calcul, nous considérons une unité de temps égale à 1s.

3.2.2.1 1^{er} Scénario : prise en compte uniquement des paquets de données

Nous ne prenons en compte que les paquets de données, donc pendant 59 secondes, le capteur est en mode réception et à la soixantième seconde il émet son paquet. La moyenne de consommation de base obtenue après ce calcul est égale à :

Consommation moyenne de base = $((24 \text{ mA} \cdot 59\text{s}) + 29\text{mA}) / 60 = 24.08 \text{ mA}$

L'étape suivante consiste à calculer la consommation moyenne en utilisant les autres modes et la comparer à la consommation moyenne de base, nous pouvons calculer par la suite la durée de vie que nous aurions pu atteindre :

- **Power Mode 1**

Consommation moyenne PM1 = $((0.2 \text{ mA} \cdot 59\text{s}) + 29\text{mA}) / 60 = 0.68 \text{ mA}$

Consommation moyenne de base / Consommation moyenne PM1 = $24.08 / 0.68 = 35.11$

En utilisant le PM1, nous pouvons atteindre une durée de vie (DDV) égale à **35 fois** la durée de vie moyenne atteinte en mode normal.

DDV PM1 = $30\text{h} \cdot 35 = 1050 \text{ heures}$

- **Power Mode 2**

Consommation moyenne PM2 = $((0.001 \text{ mA} \cdot 59\text{s}) + 29\text{mA}) / 60 = 0.48 \text{ mA}$

Consommation moyenne de base / Consommation moyenne PM2 = $24.08 / 0.48 = 50.16$

En utilisant le PM2, nous pouvons atteindre une durée de vie égale à **50 fois** la durée de vie moyenne atteinte en mode normal.

DDV PM2 = $30h * 50 = 1500$ heures

- **Power Mode 3**

Consommation moyenne PM3 = $((0.0004 \text{ mA} * 59s) + 29mA) / 60 = 0.31 \text{ mA}$

Consommation moyenne de base / Consommation moyenne PM3 = $24.08 / 0.48 = 76.78$

En utilisant le PM3, nous pouvons atteindre une durée de vie égale à **76 fois** la durée de vie moyenne atteinte en mode normal.

DDV PM3 = $30h * 76 = 2280$ heures

Nous représentons les durées de vie relatives à chaque mode dans la figure 63 :

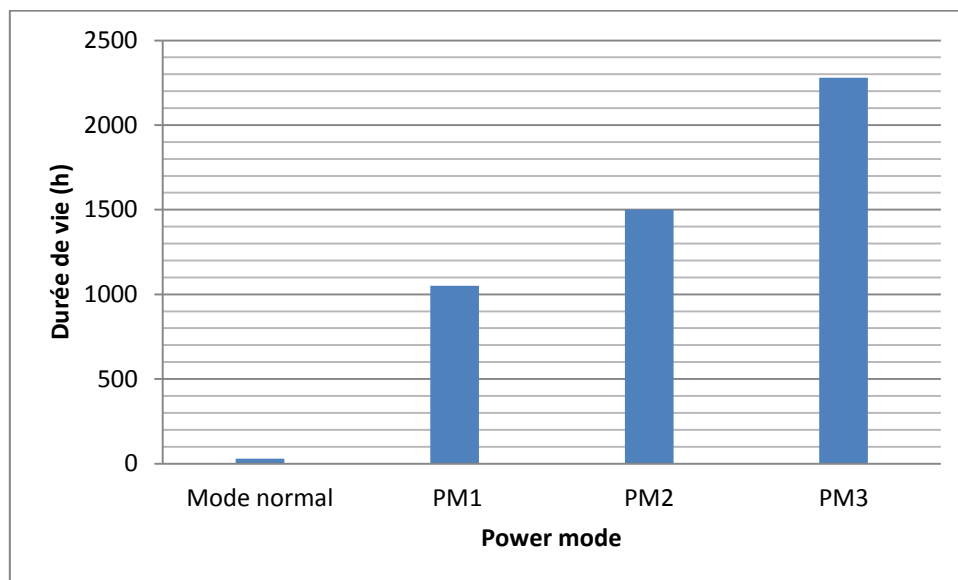


Figure 63 Durée de vie et Power mode, prise en compte que des paquets de données

En utilisant les Power Mode 1, 2 et 3, nous arrivons à une durée de vie plusieurs fois supérieure à celle atteinte en mode normal.

3.2.2.2 2^e Scénario : prise en compte des échanges réseau

Dans ce scénario, nous allons prendre en compte les échanges effectués par les capteurs afin de joindre le réseau et maintenir la connexion. Pour cela, nous avons utilisé un analyseur de trames, Wireshark, afin de déterminer l'intervalle de temps utilisé (cf. figure 64). Nous ne prenons en compte que les paquets Neighbors Solicitation (NS) émis par les capteurs et Neighbors Advertisement (NA) émis par le Nanorouteur.

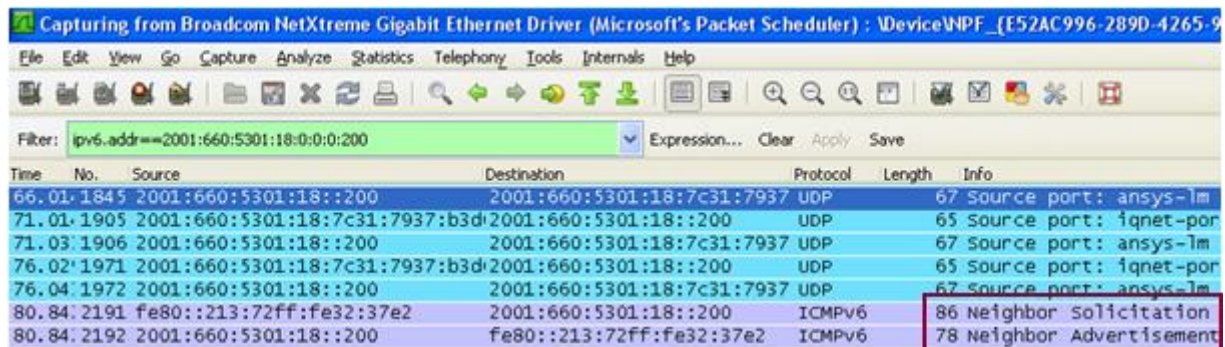


Figure 64 Wireshark Analyse des trames réseaux

D'après la capture, les trames NS et NA sont émises toutes les 42 secondes.

Nous choisissons trois cas de figures :

1^{er} cas : Maintien des échanges réseaux et échantillonnage toutes les 42s

Comme dans le premier scénario nous avons calculé la consommation moyenne dans un mode normal, la DDVM, les consommations et DDV relatives à chaque Power mode. Les résultats sont présentés dans le tableau 10.

Tableau 10 Réseau et data synchrones

Mode	Consommation (mA)	DDV (h)
Normal	24.2	30
PM1	2.09	416
PM2	1.90	463
PM3	1.90	463

D'après ce tableau, nous constatons que la DDV reste inchangeable en PM2 et PM3 parce que la différence de consommation énergétique entre les deux modes est de l'ordre de quelques micro-ampères. Néanmoins, nous réalisons un gain considérable par rapport au mode normal.

Le gain en terme de durée de vie en utilisant les Power mode 1, 2 et 3 est donné dans la figure 65.

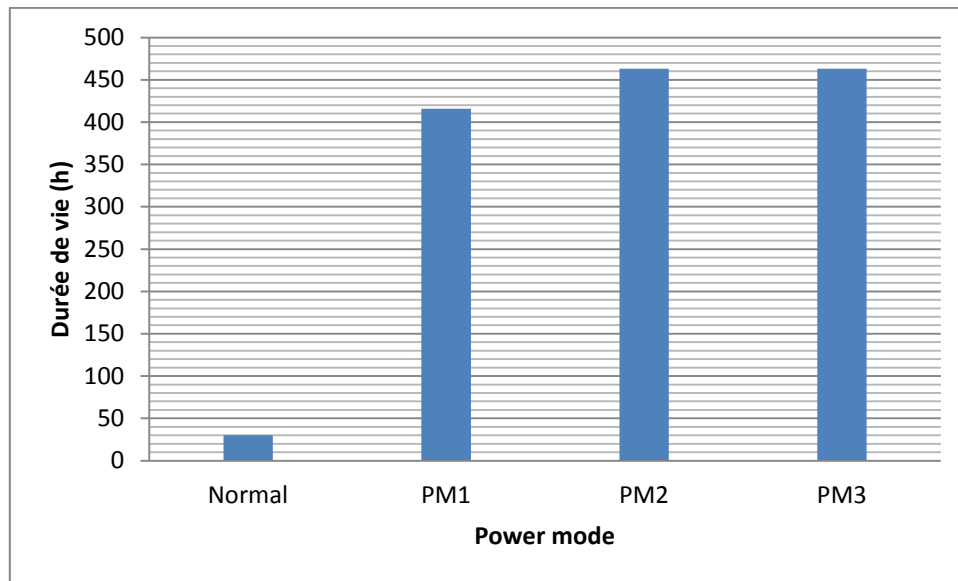


Figure 65 Durée de vie et Power mode, prise en compte que des échanges réseau

Dans ce premier cas où nous avons pris en compte les échanges réseau et une émission des paquets de données régulière, nous sommes arrivés à gagner en durée de vie en maintenant l'architecture réseau et sans perte de paquets, le niveau de service atteint est optimal. Dans les deux cas qui vont suivre, nous allons faire varier la fréquence d'échantillonnage et mesurer de nouveau le niveau de service.

2^e cas : Maintien des échanges réseaux (42s) et échantillonnage toutes les 60s

Dans ce cas, nous maintenons les échanges réseaux toutes les 42 secondes mais nous passons la fréquence d'échantillonnage à 60 secondes. Nous allons calculer dans un premier temps comme pour les autres scénarii la consommation moyenne en mode normal, la DDVM et les consommations et DDV relatives aux autres modes. Dans une deuxième étape, nous allons mesurer le délai de transmission et la gigue que nous avons constatés. Le tableau 11 illustre les valeurs calculées.

Tableau 11 Réseau et data asynchrones

Mode	Consommation (mA)	DDV (h)
Normal	24.13	30
PM1	1.5	579
PM2	1.33	653
PM3	1.33	653

D'après ce cas d'étude, nous remarquons qu'en revenant à une fréquence d'échantillonnage égale à 60s et en tenant compte des échanges des paquets NA et NS, la durée de vie est rallongée par rapport au premier cas d'étude présenté ci-dessus. En effet, nous avons moins de mesures envoyées et cela profite à la consommation d'énergie.

Par contre, en utilisant cette configuration, nous avons relevé une augmentation du délai de transmission qui est une des métriques de la qualité de service. En effet, pour réussir à

émettre son paquet, le Nanosensor doit attendre le prochain réveil. Nous avons aussi remarqué que ce délai de transmission varie car le capteur n'attend pas à chaque fois la même durée pour pouvoir émettre, ce phénomène est connu sous le nom de la gigue et est aussi une métrique de la qualité de service. Nous avons varié le taux d'endormissement du module radio et nous avons calculé à chaque fois la durée de vie, le délai de transmission et la gigue.

Nous nous basons sur un cycle égale à 42s et nous avons utilisé le PM2, le tableau 12 présente les différents éléments calculés.

Tableau 12 Délai de transmission et gigue en utilisant les Power Mode

Taux d'endormissement	Durée de vie	Délai de transmission max (s)	Gigue (s)
0,00%	30	0	
25,00%	47	25,5	13,5
50,00%	90	18	9
75,00%	150	20,5	15,78
83,00%	280	20	13,5
95,00%	416	36	15,42

Dans la figure 72, nous présentons la variation de la DDV ainsi que le délai de transmission par rapport au taux d'endormissement du module radio.

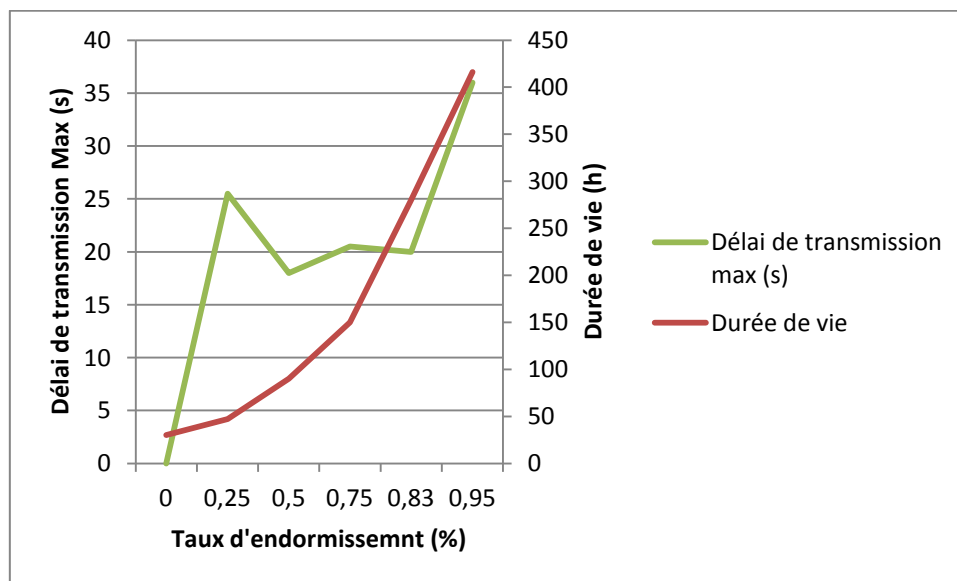


Figure 66 Sleep/Wakeup et gigue

D'après cette courbe, nous pouvons affirmer que plus le taux d'endormissement est élevé plus la durée de vie est rallongée mais, nous avons un délai de transmission assez élevé.

Grace à l'utilisation des Power mode, nous avons réussi à gagner considérablement en durée de vie des Nanosensors. Selon le scénario mis en place, le gain peut aller de 10 à 80 fois la durée de vie normale. Dans certains cas, où les échanges réseau et les paquets de données ne sont pas synchrones, nous avons relevé un impact sur deux autres métriques de la qualité de service qui sont le délai de transmission et la gigue. Il sera intéressant de mettre en place un

mécanisme de préférence dans le système c'est-à-dire selon la criticité de l'application, quelle métrique doit être privilégiée.

3.3 Conclusion

La consommation d'énergie impacte une métrique très importante dans la qualité de service des réseaux de capteurs, il s'agit de la durée de vie. Dans cette section, nous avons détaillé nos contributions afin de rallonger la durée de vie le maximum possible. La première partie de nos travaux concerne la mise en place de techniques au niveau de l'unité d'acquisition et de traitement de données. Avec ces techniques, nous avons réussi à gagner quelques heures en durée de vie. La deuxième partie concerne nos travaux au niveau de l'unité de transmission. En effectuant des simulations de mise en place de la technique sleep/wakeup, nous avons pu montrer que la durée de vie peut être rallongée jusqu'à quatre-vingt fois la durée de vie normal des Nanosensors qui ne dépasse pas les trente heures. Néanmoins, avec ce genre de mécanisme, nous avons constaté que d'autres métriques de la qualité de service peuvent être impactées en particulier le délai de transmission et la gigue.

Conclusion et perspectives

1 Conclusion générale

Les réseaux de capteurs sans fils sont devenus petit à petit des acteurs de notre vie quotidienne. Des applications multiples dans plusieurs domaines voient le jour pour améliorer notre confort et notre sécurité. Les capteurs sans fils, grâce à leurs caractéristiques, notamment leurs petites tailles, la communication sans fils et l'alimentation par batterie offrent des avantages de coûts et de facilité de déploiement. Cependant ces réseaux présentent encore des contraintes qui doivent être prises en compte dans leurs conceptions. En effet la qualité de service offerte est impactée par plusieurs facteurs liés au matériel, à l'environnement et à la consommation d'énergie. Les pannes liées souvent au matériel doivent être traitées de telle façon à permettre la continuité du service. Cette problématique a été étudiée selon diverses approches et des solutions ont été proposées telles que la redondance matérielle, la reconstruction topologique du réseau et le routage. L'énergie, cette ressource souvent limitée pour les capteurs, présente un élément majeur qui peut impacter la qualité de service. Ce facteur est associé à la métrique la plus présente dans la détermination du niveau de service, il s'agit de la durée de vie du réseau.

Dans ce travail, nous avons étudié et amélioré le fonctionnement des Nanosensors de Sensinode afin de réduire leur consommation énergétique en utilisant des techniques qui ont été présentées dans l'état de l'art de ce mémoire dans le but répondre aux questions liées à la tolérance aux pannes et la qualité de service.

Dans le chapitre 1 de ce mémoire, nous avons présenté un état de l'art sur les réseaux de capteurs sans fils. Une présentation générale de ces dispositifs, leurs caractéristiques, les domaines d'application a été faite afin de comprendre leurs modes de fonctionnement. Ensuite, nous nous sommes intéressés à l'aspect gestion des données en présentant les principaux systèmes qui peuvent être utilisés notamment les SGBD et SGFD. Les facteurs qui peuvent impacter la qualité de service ainsi que les techniques utilisées pour la maintenir ou l'améliorer ont fait l'objet de la section 3 de ce chapitre et enfin nous avons consacré la dernière section à la problématique liée à l'énergie. Nous avons présenté les différents éléments contribuant dans la dissipation de cette ressource et ensuite nous avons détaillé les solutions orientées données et réseau existantes qui peuvent être mises en place afin de conserver l'énergie.

Dans le chapitre 2, nous avons présenté le matériel Sensinode et les technologies sur lesquelles il est basé. La technologie utilisée étant 6LoWPAN, nous avons commencé par une présentation des réseaux personnels à faible portée, débit et consommation d'énergie, les LoWPAN. Les réseaux 6LoWPAN utilisent le protocole IPv6 au niveau de la couche 3 du modèle OSI, nous avons énuméré les avantages de ce « nouveau » protocole par rapport à la version « traditionnelle » qui est IPv4. Au niveau de la couche MAC, c'est le standard 802.15.4 qui est mis en place, nous avons présenté son fonctionnement et les méthodes de transfert de données utilisées relatives aux différentes topologies rencontrées. Par la suite, nous avons présenté le protocole de routage RPL, c'est celui que Sensinode a jugé compatible avec la version slotted du standard 802.15.4 qui permet l'utilisation des techniques de duty cycles. Dans la dernière section de ce chapitre, nous nous sommes

focalisés sur la composition du kit tant sur le plan matériel, les unités d'acquisition, de traitement et de transmission des données des Nanosensors N741, que sur le plan logiciel c'est-à-dire l'application NodeView 2.0 qui permet de visualiser l'état du réseau et gérer les données reçues.

Dans le chapitre 3, nous avons présenté nos contributions. Ces dernières se divisent en trois parties :

- Contributions au niveau de la gestion des données : dans cette partie, nous avons commencé par détailler les fonctionnalités des Nanosensors, l'activation des différentes sources de données et l'instauration d'un opérateur de sélection basé sur une technique de seuil et de palier. Nous avons réussi à développer et tester quatre variantes de cet opérateur. Après, nous avons présenté les modifications que nous avons apportées à l'application NodeView 2.0 afin de l'adapter à nos besoins. Nous avons réorganisé la façon de présentation des données et ajouté le facteur temps pour que chaque donnée reçue soit associée à la date et heure précise de sa réception ;
- Contributions au niveau de l'amélioration de la qualité de service : pour arriver à ce but, nous avons établi la liste des métriques à prendre en compte qui sont le délai de transmission, la gigue, le taux de perte de paquets et la durée de vie. Nous avons commencé par établir la liste des facteurs qui impactent la qualité de service dans le réseau Sensinode. En effet, les métriques citées ci-dessus peuvent être impactées par les piles utilisées, les mêmes piles peuvent donner des durées de vie différentes. La distance entre les Nanosensors et le Nanorouteur a une influence sur la qualité de service, nous avons constaté que plus la distance est grande plus le niveau de service est dégradé. Concernant la tolérance aux pannes, nous avons conclu que la meilleure technique qui peut être utilisée est la redondance matérielle ;
- Contributions au niveau de la conservation d'énergie : même si les métriques de la qualité de service sont nombreuses, la durée de vie des capteurs reste la plus importante. En effet, sans ressources énergétiques, nous ne pouvons pas parler de l'existence d'un réseau opérationnel. Nous savons, suite à l'état de l'art que nous avons présenté dans le chapitre 1, que cette ressource est dissipée à la fois par l'unité d'acquisition et de traitement et par l'unité de transmission. Nous avons réussi à rallonger la durée de vie des Nanosensors de quelques heures au niveau de l'unité d'acquisition grâce à l'utilisation des Low Power Mode, ce sont des fonctionnalités offertes par le microcontrôleur. Nous avons ensuite agi sur l'unité de transmission à deux niveaux. Le premier niveau concerne la puissance d'émission que nous avons fait varier de 1 à 20 dBm, ceci nous a permis d'atteindre un gain avoisinant une dizaine d'heures. Ensuite, nous avons simulé un mécanisme de sleep/wakeup pour voir combien de temps les batteries utilisées peuvent tenir. Nous avons montré que la durée de vie peut être prolongée à quatre-vingt fois la durée de vie moyenne qui s'élève à trente heures. Mais dans certains cas, en utilisant la technique de duty cycles, d'autres indices de qualité sont impactés comme par exemple le délai de transmission ou la gigue.

Les résultats des expérimentations que nous avons présentés dans le chapitre contributions montrent que la qualité de service offerte par le matériel Sensinode sur lequel nous avons travaillé peut être impactée par plusieurs facteurs. L'énergie est consommée assez rapidement et la durée de vie est assez limitée. Nous avons réussi, grâce, à la simulation que nous avons réalisée, à montrer que la durée de vie peut être rallongée mais pour que ce produit devient plus fiable et assure un meilleur support de la qualité de service, la technique de cycle devra être implémentée sur les Nanosensors.

Nous avons aussi apporté une contribution au projet ARTEco car les différents travaux que nous avons réalisés répondent directement aux deux motivations principales de ce projet qui sont la détection et la tolérance aux fautes et la réduction de la consommation énergétique. En apportant d'autres améliorations au comportement du matériel Sensinode au niveau de la consommation d'énergie, ce dernier pourra servir de base d'expérimentation pour ARTEco. Dans le paragraphe suivant, nous présentons deux idées qui pourront servir de point de départ pour de futurs travaux.

2 Perspectives

2.1 Gestion des données

Dans notre travail, nous avons utilisé l'application NodeView 2.0 pour la gestion des données, cette application reste limitée au matériel Sensinode. Nous avons présenté dans l'état de l'art le SGFD Sstreamware qui est un intergiciel développé au LIG qui permet de gérer les données de capteurs hétérogènes. L'idée que je propose est de développer un proxy adaptateur pour les Nanosensors afin d'offrir la possibilité de gestion des données par Sstreamware et pour remplacer NodeView 2.0. Ceci permettra d'exécuter des requêtes continues sur le flux de données généré et pouvoir modifier ces requêtes sans agir sur le code présent sur les capteurs. En plus, nous pouvons simplifier les requêtes en poussant le plus possible leurs exécutions sur les capteurs grâce à l'implémentation des opérateurs de sélection que nous avons élaborés. Nous pourrions aussi imaginer d'ajouter d'autres opérateurs comme l'agrégation.

2.2 Consommation d'énergie

Nous avons effectué plusieurs expérimentations pour mesurer et améliorer la qualité de service offerte par le réseau de capteurs Sensinode. Certes ce matériel met en évidence la technologie 6LoWPAN et nous avons noté une autoconfiguration automatique et rapide, une adaptabilité au niveau de la topologie qui dépend de la densité du réseau et de la position des capteurs, mais il présente un point faible qui est la durée de vie qui est assez limitée et qui ne pourra pas répondre aux attentes des utilisateurs.

Programmer le Soc CC2530 qui équipe le module de transmission RC2400HP de Radiocrafts s'est avéré impossible. Dans une future étape, nous pourrions travailler avec Sensinode afin de résoudre ce problème et pouvoir programmer des périodes de sommeil et d'autres de réveil. Ce point permettra de mesurer dans un environnement réel la durée de vie des Nanosensors. Dans le cas où nous aurons besoin d'effectuer des mesures toutes les heures par exemple, nous pourrions augmenter le temps de mise à jour de la topologie du réseau qui est de 42 secondes actuellement.

Annexes

1 Configuration et utilisation de IAR Embedded Workbench pour le projet MSP430

IAR Embedded Workbench est un outil de programmation spécifique pour les microcontrôleurs en particulier les produits de Texas Instrument. Plusieurs versions existent, chacune contient un ensemble de bibliothèques spécifiques à une famille de composants. Dans notre cas, le microcontrôleur qui forme le noyau des Nanosensors est le MSP430F5438 donc la version qui nous a été recommandée par Sensinode est la *IAR Embedded Workbench for MSP430 version 5.10.6 for Windows*. Nous allons présenter les étapes nécessaires à la création du projet, la configuration de l'outil et son utilisation dans le cadre de ce projet.

Etape 1 : Création de l'espace de travail (*Workspace*) et du projet

Ici, nous définissons l'espace de travail où nous allons créer notre projet, c'est là où nous retrouverons les fichiers sources, les fichiers header et le code compilé.

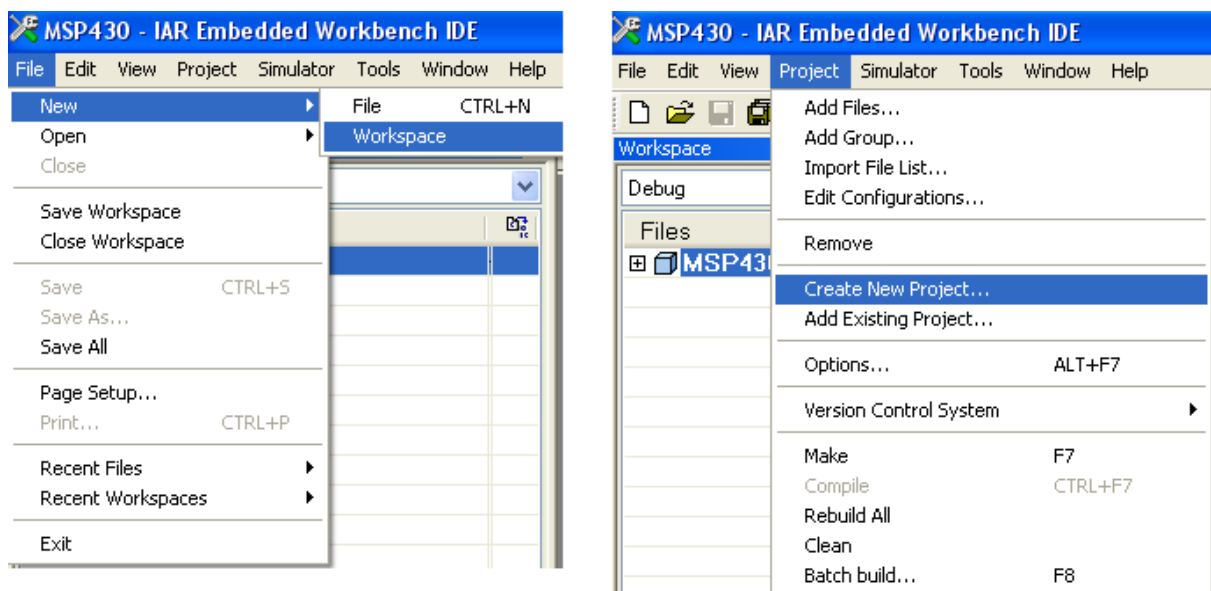


Figure 67 Création du projet

Etape 2 : Ajout des fichiers sources

Ce sont les fichiers sources qui nous ont été fournis par Sensinode. C'est le noyau de l'application qui s'exécute sur les capteurs.

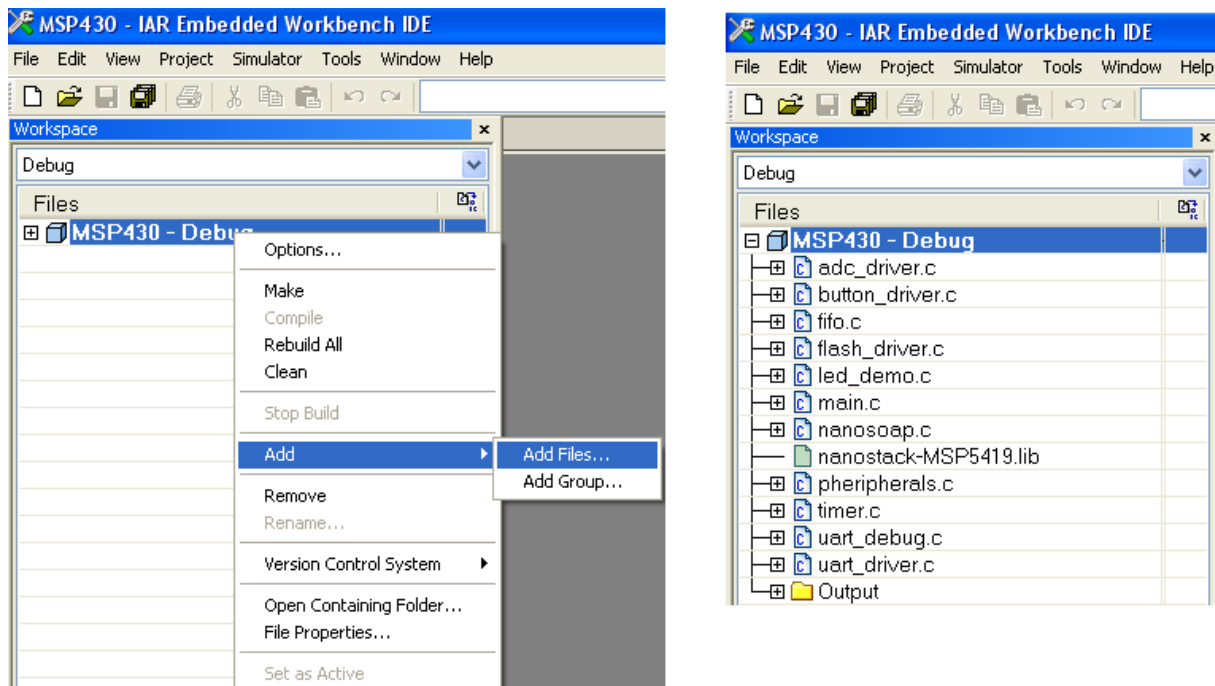


Figure 68 Ajout des fichiers sources

Etape 3 : Ajout des fichiers Header (.h)

Il s'agit d'indiquer le chemin des fichiers « .h » qui contiennent les prototypes des fonctions utilisées dans les fichiers « .c ».

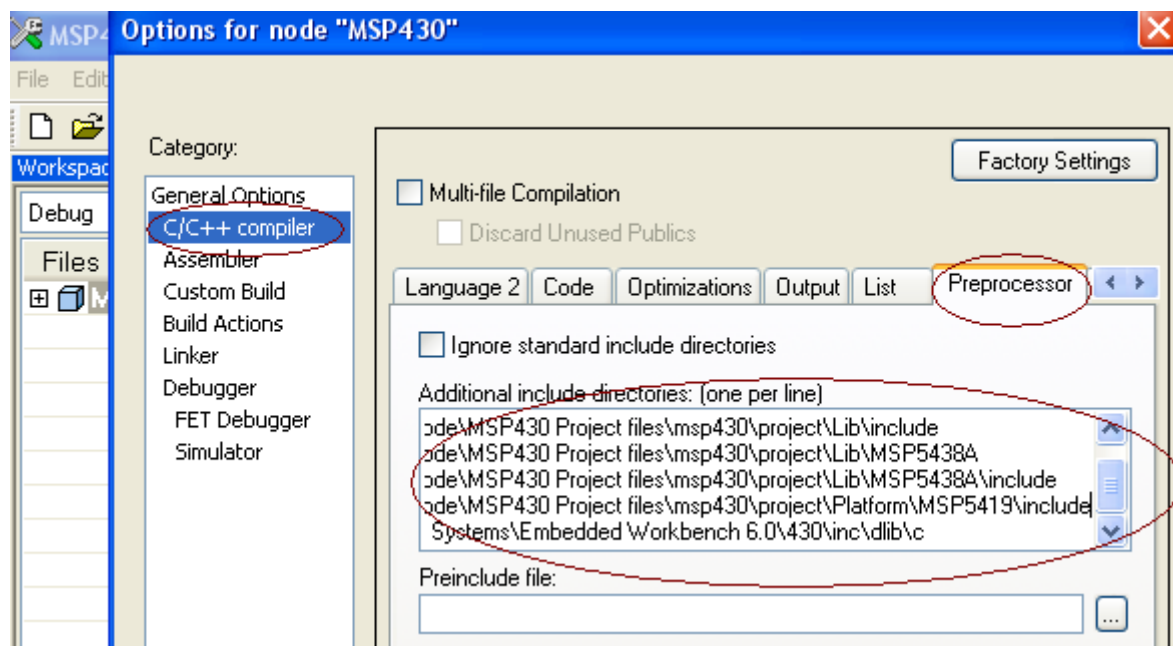


Figure 69 Ajout des fichiers .h

Etape 4 : Spécification du microcontrôleur en question

Chaque version de IAR Embedded Workbench peut être utilisée pour une famille de microcontrôleurs et le code compilé va être spécifique à une seule référence.

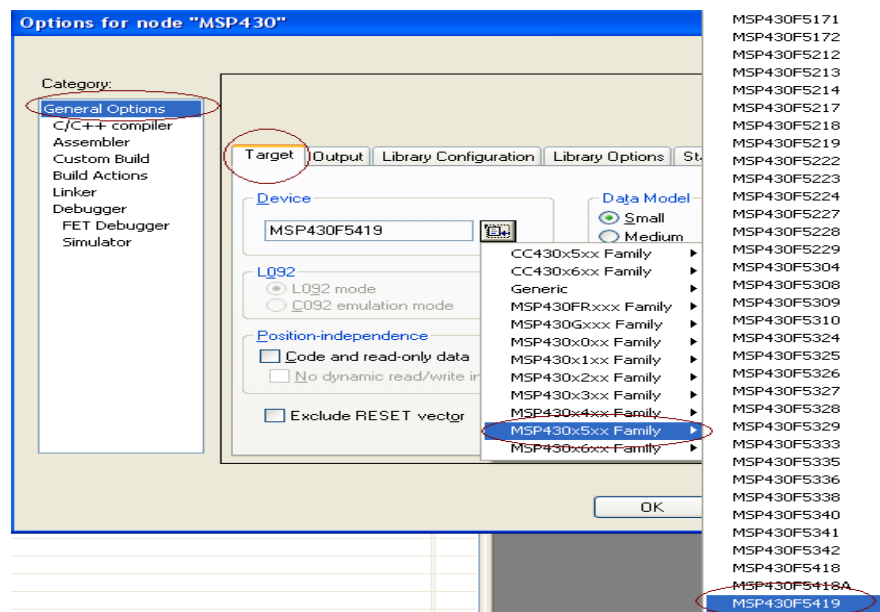


Figure 70 Choix du microcontrôleur

Etape 5 : choix du format du output

Il s'agit de définir le format de sortie du code compilé. C'est essentiel et dépend de l'outil utilisé pour flasher les capteurs. Nous choisissons un format « .txt ».

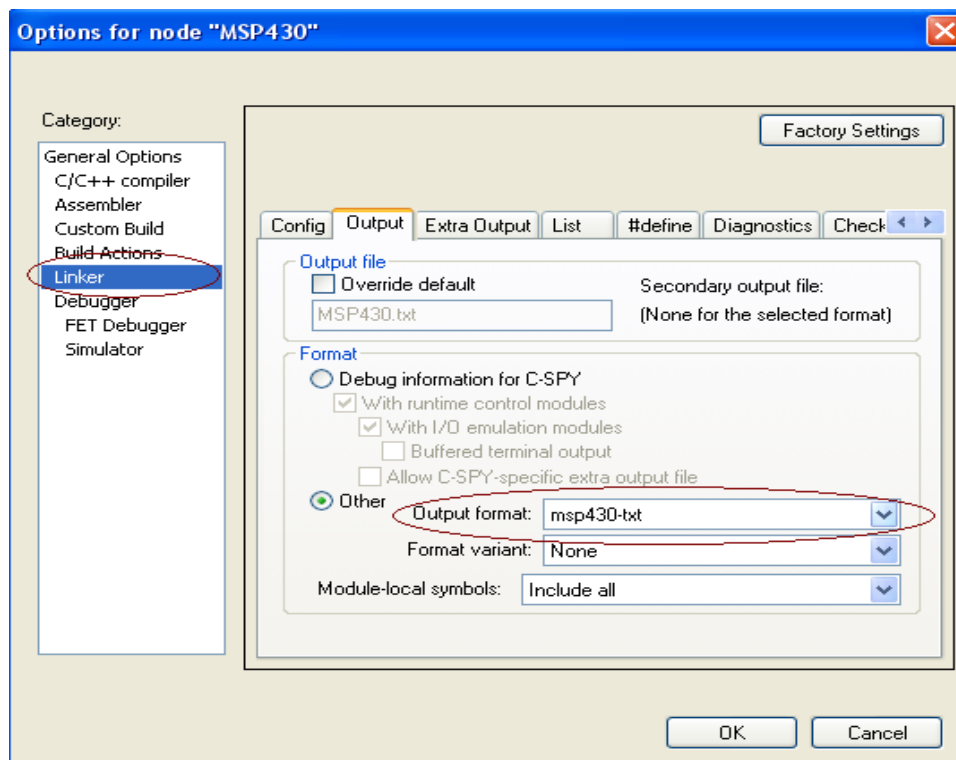


Figure 71 Format du fichier compilé

Etape 6 : Compilation du projet

Une fois le code modifié et les fonctionnalités souhaitées sont ajoutées, nous compilons le projet et la sortie est le fichier « msp430.txt ». C'est le nouveau firmware du microcontrôleur.

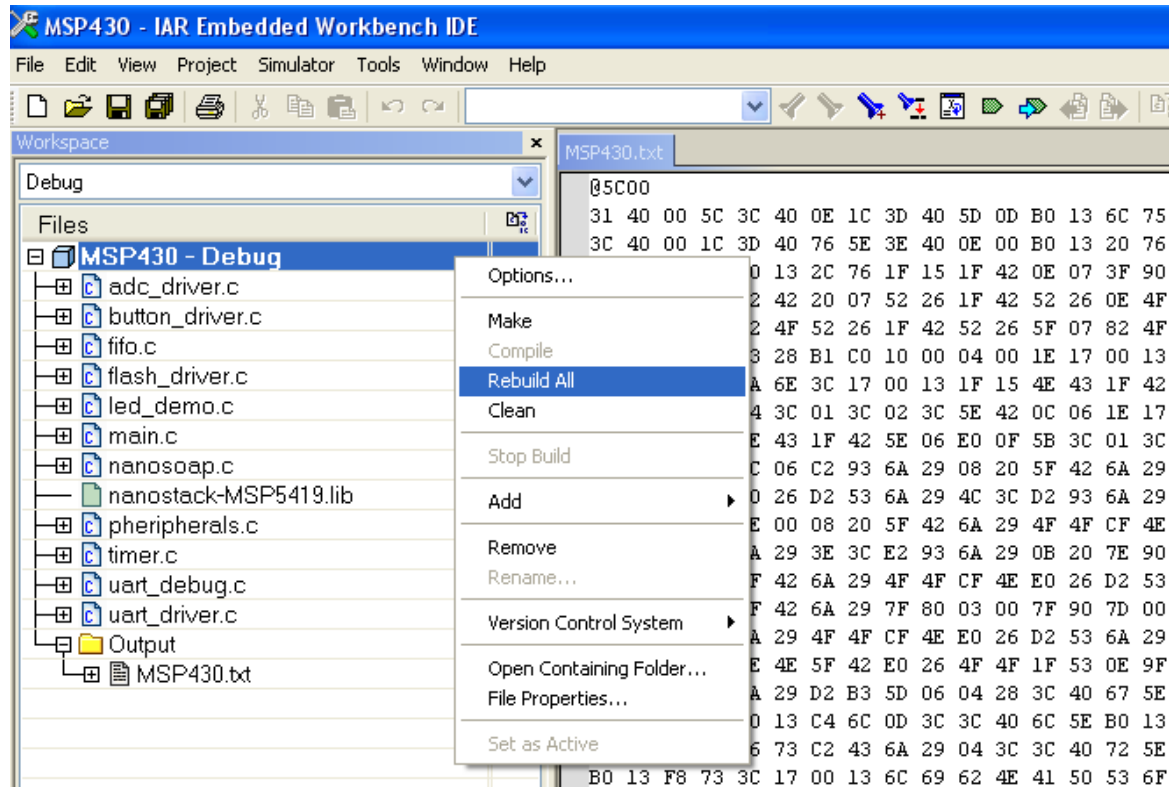


Figure 72 Compilation du projet MSP430

Etape 7 : Flasher le microcontrôleur MSP430F5438

C'est la dernière étape de la chaîne de programmation, le nouveau firmware est téléchargé sur le MSP430F5438. Cette étape est réalisée à l'aide de l'outil BSL Scripter que nous exécutons en ligne de commande.

```
C:\WINDOWS\system32\cmd.exe

E:\Mohamed\Sensinode\BSL Toolchain\BSL_Files\BSL_Scripter>BSL_Scripter.exe script.txt

-----
BSL Scripting application 1.06
The local time is: 12:04 on 21.11.2012
-----
Initializing, Mode: 5438  COM: COM8      DONE
Mass Erase:                                DONE
RX Password:                             DONE
Writing MSP430.txt to device:  DONE
CRC from 5c00 of 256 bytes                160f
Hit Unknown Command
```

Figure 73 Flasher le microcontrôleur

Bibliographie

- [1] Levent GÜRGEN, Gestion à grande échelles de données de capteurs hétérogènes. Thèse présentée le 10 septembre 2007 à Grenoble.
- [2] <http://liglab.fr> site web consulté le 12 novembre 2012.
- [3] <http://sigma.imag.fr> site web consulté le 12 novembre 2012.
- [4] Marcelo Dias de Amorim, Serge Fdida, Farid Benbadis, Luigi Iannone, Les réseaux de capteurs : Etat de l'art, exemples, recherche. Euronetlab, le 16/01/2004.
- [5] : Mohamed LEHSAINI, Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique. Thèse présentée en 2009 à Telemcen.
- [6]: Tao Shu and Marwan Krunz, Coverage-time optimization for clustered wireless sensor networks: A power-balancing approach, presented at IEEE/ACM Trans. Netw., 2010, pp.202-215.
- [7] Bilel NEFZI, Mécanismes auto-adaptatifs pour la gestion de la Qualité de Service dans les réseaux de capteurs sans fil. Thèse présentée le 21 septembre 2011, Institut National Polytechnique de Lorraine.
- [8] : Baptiste CSERNEI, Résumé Généraliste de flux de Données, Thèse présentée en février 2007, Télécom Paritech.
- [9]: L.Golab, M.T.Ozsu. Issues in Data Stream Management, ACM SIGMOD Record, June 2003.
- [10] : Annie Danzart, Sylvain Ferrandiz, Les Systèmes de Gestion de Flux de données (SGFD), Etat de l'art.
- [11]: Roncancio C. Labbé C. Bottaro A. Gurgen L. and V. Olive, SStreaMWare: a service oriented middleware for heterogeneous sensor data management. In proceeding of ICPS 2008.
- [12]: Feng Xia, QoS Challenges and Opportunities in Wireless Sensor/Actuator Networks. Volume 8 pages 1099-1110. Molecular Diversity Preservation International (MDPI), February 2008.
- [13]: Dazhi Chen and Pramod K. Varshney, QoS Support in Wireless Sensor Networks: A Survey, 2004.
- [14]: Ya Xu, Solomon Bien, Yutaka Mori, John Heidemann, Deborah Estrin, Alberto Cerpa, Topology Control Protocols to Conserve Energy in Wireless Ad Hoc Networks, January 23, 2004.
- [15] : Rahim KACIMI, Techniques de conservation d'énergie pour les réseaux de capteurs sans fils. Thèse présentée le 28 septembre 2009 à Toulouse.

- [16]: Alexander Sagen, Cyril Labbé, Mohamed Medhat Gaber, Shonali Krishnaswamy, Agustinus Borgy Waluyo, Seng Loke, A Data Clustering Approach to Energy Conservation in Wireless Sensor Networks In Proceedings of the Workshop on Ubiquitous Data Mining in conjunction with the 19th European Conference on Artificial Intelligence (ECAI 2010), August 16 - 20, 2010, Lisbon, Portugal.
- [17]: Giuseppe Anastasi, Marco Conti, Mario Di Francesco, Extending the Lifetime of Wireless Sensor Networks through Adaptive Sleep, August 2009.
- [18]: Khaled A. Arisha, Moustafa A. Youssef, Mohamed F. Younis, Energy-Aware TDMA-Based MAC for Sensor Networks. IEEE Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT 2002) New York City, New York, May 2002.
- [19] www.sensinode.com dernière consultation le 11 Novembre 2012.
- [20] P. Thubert, 6LoWPAN fragment Forwarding and Recovery. Juin 2010.
- [21] <http://www.radiocrafts.com> Dernière consultation le 19 Novembre 2012.
- [22] Roncancio C. Labbé C. Bottaro A. Gurgen L. and V. Olive, SStreaMWare: a service oriented middleware for heterogeneous sensor data management. In proceeding of ICPS 2008.
- [23]: Ranganathan Vidhyapriya and Ponnusamy Vanathi, Energy Efficient Data Compression in Wireless Sensor Networks.
- [24] <http://www.cs.virginia.edu/wsn/vigilnet/> dernière consultation le 11 Novembre 2012.
- [25] <http://www.coa.edu/greatduckisland.htm> dernière consultation le 11 Novembre 2012.
- [26] <http://standards.ieee.org> dernière consultation le 14 Novembre 2012.

MEMOIRE D'INGENIEUR C.N.A.M. en INFORMATIQUE
Evaluation de protocoles pour la Qualité de Service dans les réseaux de capteurs sans fils

Mohamed Bechir REZGUI

Grenoble, le 21 février 2013

Résumé

Depuis quelques années, les capteurs sont devenus omniprésents dans notre vie quotidienne et les applications à base de ces dispositifs ne cessent de se multiplier dans différents domaines. Ceci nous mène vers le monde de l'internet des objets où tout dispositif pourra être relié à un réseau. Dans ce type de réseau, les données sont émises en continu et leur gestion est différente selon le contexte. Dans certains cas ce sont les systèmes de gestion de base de données classiques (SGBC) qui sont privilégiés et dans d'autres ce sont les systèmes de gestion de flux de données (SGFD) qui l'emportent. Quelque soit le système utilisé, les applications exigent un niveau de service aux réseaux de capteurs. En effet les capteurs doivent faire face à plusieurs facteurs qui peuvent impacter la qualité de service notamment les pannes et des métriques sont mises en place pour évaluer la fiabilité du système. Un élément principal qui impacte la qualité de service est l'énergie. En effet cette ressource est généralement limitée et définit la durée de vie des capteurs.

La société Sensinode met en œuvre la technologie 6LoWPAN à travers son kit de capteurs Nanosensors. Nous nous sommes procurés ce matériel et nous avons étudié et amélioré son comportement sur le plan gestion des données, qualité de service et conservation d'énergie.

Mots clés : Réseaux de capteurs ; 6LoWPAN ; 802.15.4 ; données ; QoS ; pannes ; énergie.

Abstract

Since a few years, sensors become pervasive in our daily lives and applications based on these devices continue to multiply in different domains. This leads us to the world of the Internet of Things where any device can be connected to a network. In this type of network, data is transmitted continuously and managing them is different depending on the context. In some cases, database management systems (DBMS) are privileged and in others cases, data stream management system (DSMS) are used. Whatever the system used, applications require a level of service for sensor networks. In fact, sensors face several factors that can impact the quality of service, including faults. Many metrics are introduced to assess the reliability of the system. A main element which impacts the quality of service is the energy. In fact, this is generally a limited resource and defines the lifetime of sensors.

Sensinode try to implement 6LoWPAN technology through its sensor kit, Nanosensors. We have taken this material and we studied and enhance its performance on the data management, quality of service and energy conservation. All these elements are described in this document.

Keywords : Sensors networks ; 6LoWPAN ; 802.15.4 ; data ; QoS ; faults ; energy.