République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté desSciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: *Réseaux et Systèmes Distribués (R.S.D)*

Thème

Détection préventive de pannes guidée par les données dans les réseaux de capteurs sans fil

Réalisé par :

BENAHMED DAHO Amel

Présenté le 2013 devant le jury composé de MM.

- *Mr. BENMAMMAR Badr* *(Président)*
- *Mr. LEHSAINI Mohamed* *(Encadreur)*
- *Mr. BENAISSA Mohamed* *(Examinateur)*
- *Mr. BENMOUNA Youcef* *(Examinateur)*

Année universitaire: 2013-2014

Dédicaces

A ma mère, pour tant de sacrifices consentis, pour son soutien et support et tant d'amour et son courage,

A mon père, à titre posthume, ton souvenir restera à jamais gravé dans ma mémoire. Tu étais et tu resteras à jamais mon héro.

Paix à ton âme.

A ma très chère tente Fatima qui a toujours fait d'énormes efforts pour m'aider à avancer dans la vie et après le décès de mon père, paix à son âme, et jusqu'à ce jour,

A tous les membres de ma famille mes tentes, mes oncles, mes cousins et mes cousines,

A mes deux très chères et meilleures amies Soumia et Houaria pour le support et le soutient et le plein d'amour qu'elles m'ont donné durant toutes ces années d'amitié,

A toutes mes amies Aicha,Meriem Larbi,Saadia, Meriem et Assma Hadj Abdelkader, Cherifa, mon ami Sidou et mon amie Meriem Kherbouche,

A mon ami Sofiane pour son aide si précieuse, son attention et sa présence,

A mes très chers amis et partenaires de travail Mourad et Mehdi,

A mon collègue et très chère amie Téma qui m'avait soutenue avec amour et tendresse pendant ce travail,

A mon binôme Bejbouja Mohammed Amine avec qui j'ai partagé de belles années d'études , sans oublier Meriem, Amina,Ilham, Djamel, Zaq, Halima et Narymen,

A toute la promotion RSD,

A tous mes amis de la promotion SIC et tous mes amis de la promotion MID,

A tous ceux que j'aime et qui me sont chères,

Je dédie ce travail .

Résumé

Dans ce travail on a visé les applications qui nécessitent la fiabilité de l'information venue du réseau de capteurs sans fil ainsi que la fiabilité des nœuds eux même pour garantir le bon fonctionnement du réseau mais aussi une prévention de pannes et un recouvrement d'erreur afin d'éviter tout accident ou incendies que peut subir un RCSF. Pour cela on a eu recours à des méthodes d'analyse de données, qui auront pour rôle de traiter les données reçues par les capteurs suivant des calcules statistiques qui vont déterminer les anomalies et les défaillances au niveau des capteurs qui peuvent présenter des pannes dans le futur avant que celles-ci se produisent et par suite corriger cela avant tout incident.

La mise en place de cette approche exige des outils matériels et logiciels bien spécifiques puisque les capteurs sont des composants à ressources limitées. Pour cela nous avons utilisé le langage NesC qui fait minimiser l'utilisation de la mémoire et TinyOs qui est un système d'exploitation léger.

Mots clés : Réseaux de capteurs, Pannes, Fautes, parallélisme, tolérance, détection, agrégation, clustering, NesC, TinyOs.

Abstract

In this thesis we aimed at the applications that require the reliability of the information coming from the network of wireless sensors as well as the reliability of the motes to guarantee the smooth running of the network but also a prevention of breakdowns and a covering of error to avoid any accident or set on fire that a RCSF can undergo. For it we resorted to methods of analysis of data, which will have for role to process the data received by the sensors according to statistics which are going to determine the anomalies and the failures at the level of the sensors who can present breakdowns in the future before these occur and as a consequence to correct it before any incident.

The implementation of this approach requires very specific hardware and software tools since the sensors are limited resources components. For this we used the NESC language that minimize the use of memory and TinyOS is a lightweight operating system.

Keywords: Sensor networks, Failures, Faults, parallelism tolerance, detection, aggregation, clustering, NESC TinyOS.

Remerciements

Soyons reconnaissants aux personnes qui nous donnent du bonheur ; elles sont les charmants jardiniers par qui nos âmes sont fleuries.

Marcel Proust (1871-1922).

Merci !

Je voudrais adresser mes vifs remerciements à mon promoteur le Dr Mohammed Lehsaini qui a accepté de me prendre par la main pour faire ce chemin de questionnement jusqu'à son aboutissement.

Pour les nombreux sacrifices qu'il a consentis à l'aboutissement du présent travail qu'il a suivi patiemment, avec doigté et surtout avec le professionnalisme qu'on lui connaît et également pour tous les partages scientifiques et du savoir pour la vie de tous les jours, je lui rends le témoignage de toute ma reconnaissance.

J'adresse mes remerciements les plus sincères à toutes les personnes qui m'ont aidée et ont contribué à l'élaboration de ce mémoire.

Je tiens également à remercier Mr. *BENMAMMAR* pour l'honneur qu'il m'a fait de présider le jury de soutenance et je lui exprime ma profonde gratitude ainsi qu'aux membres du jury Mr. *BENMOUNA* et Mr. *BENAISSA* qui m'ont consacrée une partie de leur temps précieux.

Enfin, je n'oublierai pas aussi mes proches et amis qui m'ont vivement soutenue et encouragée au cours de la réalisation de ce mémoire.
Et … à ma très chère maman, pour sa bénédiction, sans son soutien moral et sa patience, un grand merci.

Merci à vous tous.

Table des matières

Liste des figures

Liste des Tableaux

Introduction générale

Introduction générale

L'évolution rapide de la technologie dans le domaine de la communication sans fil et de la micro-électronique, a donné naissance à des équipements miniaturisés dotés d'une unité de calcul, de composants pour la collecte de l'information, et d'une antenne pour transmettre l'information à un centre de contrôle distant. Ces équipements sont appelés nœuds capteurs ou « motes ». Ils ont la capacité de s'auto-organiser pour former un réseau appelé réseau de capteurs. Ces équipements sont généralement déployés dans des environnements hostiles pour surveiller ou collecter de l'information dans un champ de captage. Ils présentent une autonomie d'énergie puisqu'ils sont dotés d'une source d'énergie limitée (batterie) qui est généralement non rechargeable et difficile à la remplacer. [1]

En outre, ces nœuds sont sujets à des pannes pour différentes causes (épuisement de l'énergie, écrasement par des animaux, etc.…). Dans ce cas de figure, il y aurait un grand risque que le réseau perd l'aspect de connexité et que l'information ne peut pas être transmise au centre de contrôle ou cette information est erronée à cause de la faille des capteurs. Pour faire face à ce type de scénarios, les capteurs sont généralement déployés en grand nombre et ils ont la capacité de s'auto-organiser. Au début, un nombre minimal de capteurs est impliqué dans la formation des réseaux et les autres passent au mode veille pour préserver leurs batteries ce qui permet de prolonger la durée de vie du réseau. Quand un capteur cesse de fonctionner et le réseau perd sa connexité, un autre capteur se trouvant dans son voisinage passe au mode actif pour le remplacer de telle sorte que le réseau soit toujours connexe. Or, dans d'autres cas un capteur peut perdre sa fiabilité et l'information qu'il remonte, est erronée. Dans certaines applications, cette information peut être importante par exemple quand il s'agit de la surveillance d'un patient. De ce fait, la non-fiabilité de l'information peut provoquer un autre traitement qui aura une conséquence négative sur la santé du patient. Dans cette optique, on propose de mettre en place une plateforme qui permet de détecter les capteurs défaillants. En outre, on propose d'analyser les données collectées après chaque période et détecter les capteurs défaillants dans une étape précoce pour éviter toute conséquence négative surtout dans les applications sensibles à l'instar des applications médicales et militaires.

Les réseaux de capteurs sans fil font partis des nouvelles technologies qui ont bouleversé le monde et notre manière de vivre et de travailler. Ils répondent à l'émergence ces dernières décennies, de l'offre et d'un besoin accru d'observation et de contrôler des phénomènes physiques et biologiques dans différents domaines. Ils sont

généralement composés d'un grand nombre d'unités de traitement, appelées « motes », communiquant via des liens sans fil. Les différences entre les réseaux traditionnels (réseaux télécommunication, Internet, etc.) et les réseaux de capteurs sont les suivantes :

- Les réseaux de capteurs ont plus de nœuds avec une plus haute densité
- Les nœuds dans les réseaux de capteurs sont assez fragiles et vulnérables à diverses formes de défaillances : cassure, faible énergie, etc.

Dans ce travail on étudie les différentes causes de pannes qui peuvent survenir durant la vie d'un réseau de capteurs sans fil. Puis on a instauré un mécanisme qui permet de détecter les pannes dans une étape précoce dans un réseau de capteurs. Cette détection préventive de pannes permet de faire appel au processus d'auto-organisation avant que la panne aura une conséquence fatale sur le fonctionnement du système.

Pour se faire, on a organisé notre travail autour de trois chapitres encadrés par une introduction et une conclusion :

Le chapitre 1: est une introduction aux réseaux de capteurs sans fil, une présentation de leur fonctionnement général, les applications potentielles et les principales caractéristiques.

Le chapitre 2 : Est une présentation des différentes catégories de pannes qui peuvent survenir durant le cycle de vie d'un réseau de capteurs sans fil, leurs caractéristiques et causes ainsi que les approches et solutions proposées pour la tolérance aux fautes.

Le chapitre 3 : constitue le cœur de notre travail. Dans ce chapitre, on propose l'instauration d'un mécanisme de tolérance aux fautes dans un réseau de capteurs suivant une méthode d'analyse de données et de recouvrement d'erreur préventive. L'architecture de déploiement des nœuds dans ce réseau est plate c'est-à-dire que tous les nœuds ont les mêmes caractéristiques.

Enfin, on conclue ce mémoire par une conclusion générale et on présente quelques perspectives.

Chapitre 1

Concepts généraux sur les réseaux de capteurs

Chapitre 1 Concepts généraux sur les réseaux de capteurs

I.1 Introduction

L'évolution technologique a été marquée par la miniaturisation des équipements. Ces dernières années, il y avait l'apparition des dispositifs à bas coûts intégrant les fonctionnalités de captage, de traitement et de communication. Ces dispositifs sont déployés dans la nature afin de créer un réseau de capteurs à des fins aussi bien de contrôle que de monitorisation. Ils permettent de détecter les changements environnementaux, biologiques, etc.… et les signalent aux autres nœuds selon une architecture flexible du réseau. Les nœuds capteurs donnent la possibilité d'être déployés dans des environnements hostiles où l'intervention humaine est quasiment impossible.

I.2 Les réseaux de capteurs sans fil

Un réseau de capteurs sans fil (RCSF) est un réseau ad hoc avec un grand nombre de nœuds qui sont des micro-capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome. La position de ces nœuds n'est pas obligatoirement prédéterminée. Ils peuvent être aléatoirement dispersés dans une zone géographique, appelée « champ de captage » correspondant au terrain d'intérêt pour le phénomène capté (ex : largage des capteurs par un avion sur un volcan pour étudier les phénomènes vulcanologiques et leurs évolutions, etc...) [2].

I.2.1 Génération des réseaux de capteurs

Les réseaux de capteurs sans fil (RCSF) ont montré leur impact sur notre vie quotidienne. Dans [3], les auteurs ont classé le processus de création de capteurs en trois générations. Le tableau I-1 illustre cette catégorisation des capteurs en des générations.

Génération	Période	Taille	Poids	Batterie
1er	Les années 80 et 90	Grande boîte à chaussures	Kilogrammes	Grosse
2e	2000-2003	Boîte de cartes	Grammes	AA
3e	2010	Particule de poussière	Négligeable	Solaire

Tableau I-1: Génération des réseaux de capteurs

I.2.2 Le nœud capteur

Un nœud capteur est composé de plusieurs modules dont chacun d'eux a une tâche particulière : acquisition, traitement, ou transmission de données. Il comprend également une source d'énergie [4] :

- **L'unité d'acquisition** : l'unité d'acquisition est composée d'un ou plusieurs capteurs qui effectuent des mesures numériques sur les paramètres environnementaux et d'un Convertisseur Analogique/Numérique (CAN) qui convertit l'information relevée et la transmettre à l'unité de traitement.
- **L'unité de traitement :** l'unité de traitement est composée de deux interfaces : une interface pour l'unité d'acquisition et une interface pour l'unité de transmission. Cette unité est également composée d'un processeur et d'un système d'exploitation spécifique. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de transmission.
- **L'unité de transmission :** cette unité est responsable de toutes les émissions et réceptions de données via un support de communication radio.
- **La source d'énergie :** Pour alimenter les unités citées ci-dessus, un capteur dispose d'une source d'énergie qui est généralement des piles de type AA.

La figure I.1 [5] illustre l'architecture d'un nœud capteur.

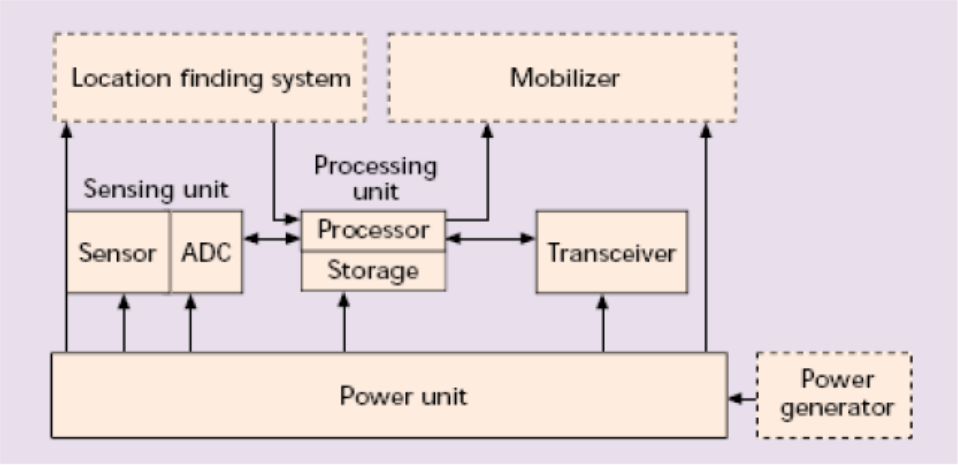

Figure I-1: Composants d'un nœud capteur

I.2.3 Type de capteurs

Il existe plusieurs types de capteurs, avec des fonctionnalités diverses et variées. La plupart des capteurs dépendent de l'application pour lesquels ils ont été conçus (capteurs aquatiques, sous-terrain, etc. . .). La figure I.2 [5] illustre l'évolution des capteurs au cours de ces 20 dernières années. Cette représentation met en avant l'importance des travaux de recherche de l'université de Berkeley dans l'essor des

réseaux de capteurs, surtout sachant que l'entreprise Xbow2 (aussi appelé Crossbow) qui fait jusqu'à aujourd'hui office de référence dans la fabrication de capteurs.

Les capteurs fabriqués par Xbow au cours des dix dernières années (famille de capteurs Mica et Telos) sont sans aucun doute les plus utilisés dans les expériences et travaux de recherche. Ces capteurs sont capables de mesurer plusieurs métriques et utilisent Chipcon CC2420 qui est devenu le standard au niveau des modules de transmission utilisant le protocole de communication IEEE 802.15..4 [2].

Figure I-2: Evolution des capteurs

I.2.4 Architecture des réseaux de capteurs

Un réseau de capteurs est constitué de milliers de nœuds appelés nœuds capteurs ou tout simplement capteurs, permettant de capter et collecter des événements, d'analyser les traitements et de transmettre les informations recueillies dans différents environnements. Ces nœuds peuvent avoir des positions fixes ou bien être déployés aléatoirement pour surveiller l'environnement. Les communications dans un réseau de capteurs se font souvent d'une manière multisaut jusqu'à la station de base qui est un nœud possédant plus de ressources matérielles et permettant de collecter et stocker les informations issues des capteurs. En d'autres termes le fonctionnement d'un réseau de capteurs se déroule de la manière suivante : les nœuds sont déployés dans une zone appelée zone d'intérêt pour la surveiller. Lorsqu'un nœud détecte un évènement, il le traite localement et l'achemine vers la station de base via une communication multi-saut. Ce processus est illustré dans la figure 1.3[5].

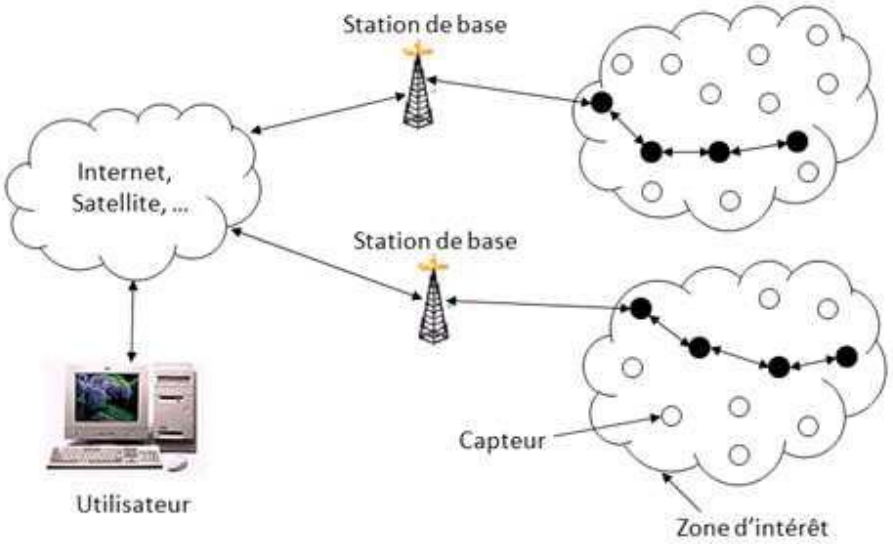

Figure I-3: Exemple d'un réseau de capteurs

Il existe plusieurs topologies pour les réseaux de capteurs : [3]

- **Topologie en étoile :** La topologie en étoile est un système uni-saut. Tous les nœuds envoient et reçoivent seulement des données avec la station de base. Cette topologie est simple et elle demande une faible consommation d'énergie. Elle est recommandée quand la distance entre les nœuds et la station est limitée.
- **Topologie en toile (Mesh Network) :** La topologie en toile est un système multi-saut dont lequel la communication entre les nœuds et la station de base est possible. Chaque nœud a plusieurs chemins pour envoyer des données. Cette topologie a plus de possibilités de passer à l'échelle du réseau, avec redondance et tolérance aux fautes, mais elle demande une consommation d'énergie plus importante.
- **Topologie hybride :** La topologie hybride est un mélange des deux topologies ci-dessus. Les stations de base forment une topologie en toile et les nœuds autour d'elles sont en topologie étoile. Elle assure la minimisation d'énergie dans les réseaux de capteurs. La figure I.4 [5] illustre cette topologie.

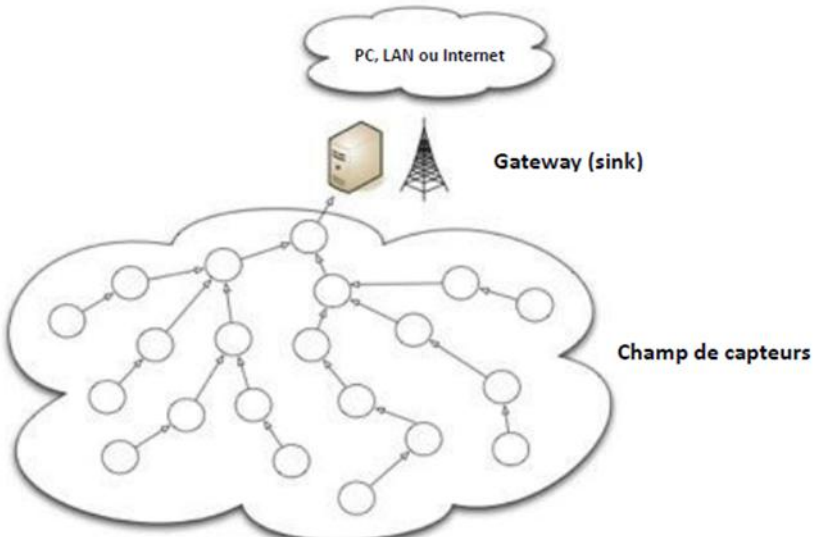

Figure I-4: Topologie hybride d'un RCSF

I.3 Applications

La diminution de taille et de coût des micro-capteurs, l'élargissement de la gamme des types de capteurs disponibles (thermique, optique, vibrations...) et l'évolution des supports de communication sans fil, ont élargi le champ d'applications des réseaux de capteurs. Parmi elles, on cite [6] :

- **Applications militaires :** On peut penser à un réseau de capteurs déployé sur un endroit stratégique afin de surveiller toutes les activités des forces ennemies, ou d'analyser le terrain avant d'y envoyer des troupes (détection d'agents chimiques, biologiques ou de radiations).
- **Applications domestiques :** En plaçant, sur le plafond ou dans le mur, des capteurs, on peut économiser l'énergie en gérant l'éclairage ou le chauffage en fonction de la localisation des personnes.
- **Applications environnementales :** Les réseaux de capteurs sont beaucoup appliqués dans ce domaine pour détecter des incendies, surveiller des catastrophes naturelles, détecter des pollutions et suivre des écosystèmes.
- **Applications agricoles :** Dans les champs agricoles, les capteurs peuvent être semés avec les graines. Ainsi, les zones sèches seront facilement identifiées et l'irrigation sera donc plus efficace et économique.
- **Applications médicales :** Les réseaux de capteurs ont aussi des développements dans le domaine de diagnostic médical. Par exemple, des micro-caméras sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain avec une autonomie de 24 heures.

- **Applications de transport :** Il est possible d'intégrer des nœuds capteurs au processus de stockage et de livraison. Le réseau ainsi formé, pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison.

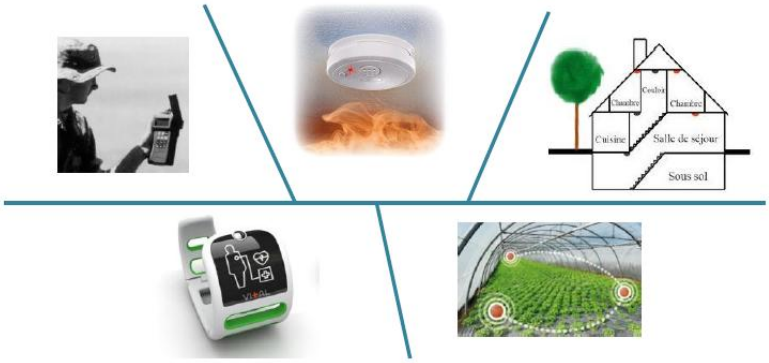

Figure I-5 : Applications des réseaux de capteurs

I.4 Technologies de communication dans les RCSF

- **Bluetooth / IEEE 802.15.4 :** Bluetooth est une spécification de l'industrie des télécommunications. Elle utilise une technique radio courte distance destinée à simplifier les connexions entre les appareils électroniques. Malheureusement, un grand défaut de cette technologie est sa trop grande consommation d'énergie [5].
- **ZigBee / IEEE 802.15.4 :** ZigBee est une norme de transmission de données sans fil permettant la communication de machine à machine. Zigbee offre des débits de données moindres, mais sa très faible consommation électrique et ses coûts de production très bas ont fait d'elle une candidate idéale pour les équipements à ressources limitées.
- **Dash 7 / ISO/IEC 18000-7 :**Dash7 est une nouvelle technologie de communication utilisant la norme ISO/IEC 18000-7. Sa consommation électrique est très faible, la durée de vie de batterie peut arriver à plusieurs années et sa portée est 2km en outdoor. Elle fournit une faible latence pour le suivi des objets en mouvement et un débit de transmission allant jusqu'à 200kbits/s.

I.5 Les différents facteurs de conception

La conception des réseaux de capteurs est influencée par de nombreux facteurs comme la tolérance aux pannes, les coûts de déploiement, la consommation d'énergie, l'environnement ou la topologie du réseau. Ces facteurs représentent la base de la conception de protocoles ou d'algorithmes pour les réseaux de capteurs.

I.5.1 Tolérance aux pannes

Les nœuds peuvent être sujets à des pannes dues à leur fabrication (ce sont des produits de série bon marché, il peut donc y avoir des capteurs défectueux) ou plus

fréquemment à un manque d'énergie. En outre, les interactions externes (chocs, interférences,. .) peuvent aussi être la cause de leur dysfonctionnement. De ce fait, afin que les pannes n'affectent pas la tâche première du réseau, il faut mettre en place des mécanismes qui permettent d'augmenter le taux de disponibilité d'un réseau de capteurs [5].

I.5.2 Topologie du réseau

En raison de leur forte densité dans la zone à surveiller, il faut que les nœuds capteurs soient capables d'adapter leur fonctionnement afin de maintenir la topologie souhaitée [7].On distingue généralement trois phases dans la mise en place et l'évolution d'un réseau:

- **Déploiement :** Les nœuds sont soit répartis de manière prédéfinie soit de manière aléatoire (largués en masse depuis un avion). Il faut alors que ceux-ci s'organisent de manière autonome.
- **Post-Déploiement - Exploitation :** Durant la phase d'exploitation, la topologie du réseau peut être soumise à des changements dus à des modifications de la position des nœuds ou bien à des pannes.
- **Redéploiement :** L'ajout de nouveaux capteurs dans un réseau existant implique aussi une remise à jour de la topologie.

I.5.3 Consommation d'énergie

L'économie d'énergie est une des problématiques majeures dans les réseaux de capteurs. En effet, la recharge des sources d'énergie est souvent trop coûteuse et parfois impossible. Il faut donc que les capteurs économisent au maximum l'énergie afin de pouvoir fonctionner et garantir une longue durée de vie.[8]

I.6 Caractéristiques des RCSF

Les principales caractéristiques des réseaux de capteurs se résument dans ce qui suit :

- **Densité importante des nœuds :** Les réseaux de capteurs se composent généralement d'un nombre très important des nœuds pour garantir une couverture totale de la zone surveillée. Ceci engendre un niveau de surveillance élevé et assure une transmission plus fiable des données.
- **Topologie dynamique :** L'instabilité de la topologie des réseaux de capteurs est le résultat des trois facteurs essentiels :
  - **La mobilité des nœuds :** les nœuds capteurs peuvent être attachés à des objets mobiles qui se déplacent librement et arbitrairement, introduisant ainsi une topologie instable du réseau.

- **La défaillance des nœuds :** du fait de l'autonomie énergétique limitée des nœuds, la topologie du réseau n'est pas fixée (les nœuds qui épuisent leur énergie, sont considérés comme des nœuds inexistants).
- **L'ajout de nouveaux nœuds :** de nouveaux nœuds peuvent facilement être rajoutés. Il suffit de placer un nouveau capteur qui soit dans la portée de communication d'au moins d'un autre nœud capteur du réseau déjà existant.

- **Auto-organisation :** L'auto organisation s'avère très nécessaire pour ce type de réseaux afin de garantir sa maintenance. Vu les différentes conséquences résultant de l'instabilité de la topologie du réseau de capteurs, ce dernier devra être capable de s'auto-organiser pour continuer ses applications.
- **Scalabilité :** Les réseaux de capteurs peuvent contenir des centaines voire des milliers de nœuds capteurs [7]. Un nombre aussi important engendre beaucoup de transmissions internodales et nécessite que le nœud « Sink » soit équipé d'une mémoire importante pour stocker les formations reçues.

I.7 Conclusion

Dans ce chapitre, on a fait une présentation des réseaux de capteurs, leurs caractéristiques et leurs domaines d'applications. Cette étude nous permet de poser les briques de base et fédérer quelques concepts nécessaires à la compréhension de la problématique. La flexibilité, la tolérance aux fautes, le prix réduit et le déploiement rapide des réseaux de capteurs offrent des possibilités infinies de développement dans tous les domaines d'applications. Ceci permet de penser que les réseaux de capteurs feront bientôt partie intégrante de notre vie et satisferont sûrement les plus grands projets.

Dans le chapitre suivant, on aborde les causes de pannes que peut subir un réseau de capteurs ainsi que les approches de tolérance aux pannes et aux fautes dans les RCSF.

Chapitre2

Tolérance aux pannes dans les réseaux de capteurs

Chapitre 2 Tolérance aux pannes dans les réseaux de capteurs

II.1 Introduction

Certains capteurs peuvent être bloqués ou tomber en panne à cause d'un manque d'énergie, d'un dégât matériel ou d'une interférence environnementale. La panne d'un capteur ne doit pas affecter le fonctionnement global de son réseau. C'est le problème de fiabilité ou de tolérance aux pannes. La tolérance aux pannes a pour objectif de maintenir les fonctionnalités du réseau sans interruption due à une panne de certains capteurs.

II.2 Les pannes dans les réseaux de capteurs

II.2.1 Notion de panne (faute, erreur)

Dans cette section, on présente quelques notions liées à la tolérance aux fautes dans les RCSF :

- **La défaillance** survient quand le système a un comportement anormal : une erreur est la partie de l'état du système (par rapport au processus de traitement) qui est susceptible d'entraîner une défaillance. La cause supposée de l'erreur est une faute. Une erreur est donc la manifestation d'une faute dans le système, et une défaillance est donc l'effet d'une erreur sur le service.
- **Une faute active** lorsqu'elle produit une erreur. Elle pourrait être soit une faute interne qui était précédemment dormante c'est-à-dire qu'elle ne produisait pas d'erreur et qui a été activée par le processus de traitement, soit une faute externe [9].
- **Une faute interne** peut passer, de manière cyclique, de l'état dormant à l'état actif. Une erreur est, par nature, temporaire. Elle peut être latente ou détectée: une erreur est latente tant qu'elle n'a pas été reconnue en tant que telle; elle est détectée soit par des mécanismes de détection d'erreur qui analysent l'état du système, soit par l'effet de l'erreur sur le service (défaillance).

Généralement, une erreur propage d'autres erreurs, nouvelles, dans d'autres parties du système. Une défaillance survient lorsqu'une erreur traverse l'interface système-utilisateur et affecte le service délivré par le système. Si un système peut être considéré comme un ensemble de composants, la conséquence de la défaillance d'un composant est une faute interne pour le système qui le contient, et aussi une faute externe pour le ou les composants qui interagissent avec lui. Ceci conduit à la chaîne fondamentale suivante :

... →défaillance → faute →erreur →défaillance →faute →...

II.2.2 Classification des pannes

Les pannes qui peuvent survenir dans un réseau de capteurs sans fil, sont classées selon trois critères : durée, cause et comportement. La classification classique est faite selon la nature résultante de la panne ce qui donne deux types de pannes : [10]

- **Pannes catastrophiques** : elles sont inacceptables. Par exemple, les grandeurs pathologiques d'un patient.
- **Pannes non catastrophiques** : elles sont acceptables. La collecte de la température dans un environnement.

La deuxième classification qui suit les trois critères est illustrée dans la figure suivante (Figure II-1) [10]:

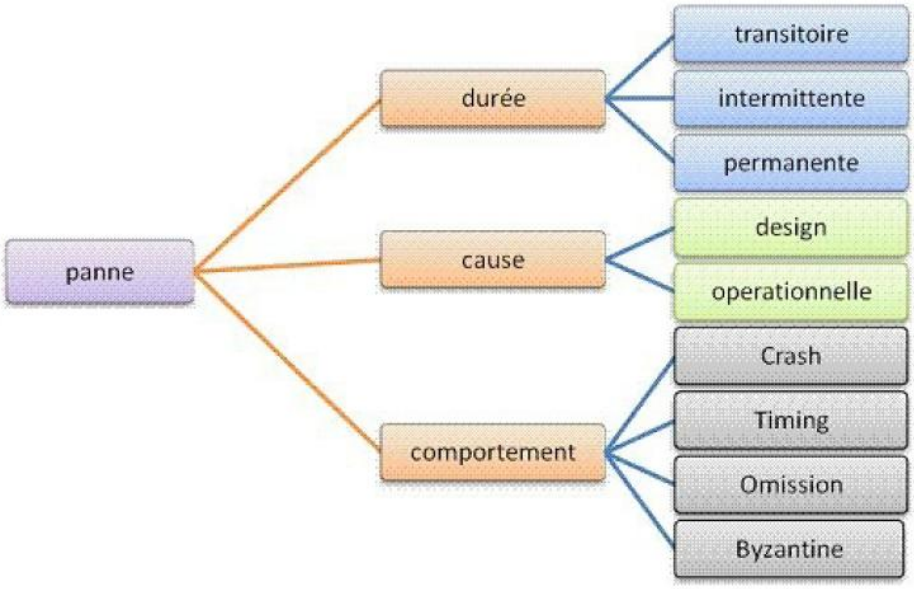

Figure II-1: Classification des pannes

II.2.3 Causes de pannes

Les capteurs peuvent subir des pannes et des défaillances dues à de différentes causes et phénomènes qui peuvent être internes ou externes selon leurs origines. Ces pannes que peut subir un réseau de capteurs peuvent être dues à : [11]

- L'épuisement d'énergie des capteurs.
- Perte de connexion sans fil due à l'épuisement de la batterie d'un capteur.
- Destruction physique (accidentelle ou volontaire).
- Interférences environnementales.

II.3 La tolérance aux pannes

La limitation d'énergie dans les capteurs sans fil, et les environnements hostiles dans lesquels ils pourraient être déployés, sont des facteurs qui rendent ce type de réseaux non fonctionnels dans certaines périodes. Ainsi, la perte de connexions sans fil peut être due à une extinction d'un capteur suite à un épuisement de sa batterie, ou tout simplement à une destruction physique accidentelle ou intentionnelle par un ennemi. Par ailleurs, la non fiabilité de type de capteurs est une caractéristique qui augmente les risques de pannes sur ce type de réseau.

Etant donné que les réseaux de capteurs reposent sur des protocoles de communications ad hoc, il est donc nécessaire de considérer la tolérance aux pannes comme critère indispensable dans la conception de ces protocoles [12]. En outre, afin d'assurer la communication entre le nœud collecteur et les autres nœuds d'un réseau de capteurs, les protocoles de routage sont basés sur la communication multisauts dans laquelle chaque nœud joue alors, en plus du rôle de source de données, le rôle d'un routeur. Toutefois, ces nœuds sont sujets à de nombreuses pannes, dues principalement à l'épuisement des batteries et aux destructions physiques (par exemple, suite à un écrasement par des animaux). Ainsi, la panne des nœuds entraîne la perte des liens de communications et donc un changement significatif dans la topologie globale du réseau. Ceci peut affecter d'une façon considérable la connectivité du réseau et diminuer, en conséquence, sa durée de vie.

II.3.1 Définition

La propriété de tolérance aux pannes est définie par l'aptitude du réseau à maintenir ses fonctionnalités, en cas de panne de certains de ses nœuds. Elle vise donc à minimiser l'influence de ces pannes sur la tâche globale du réseau [10].

II.3.2 Procédure générale de tolérance aux pannes

La conception d'une procédure pour la tolérance aux pannes dépend de l'architecture et des fonctionnalités du système. Cependant, certaines étapes générales sont exécutées dans la plupart des systèmes comme s'est illustré dans la figure :

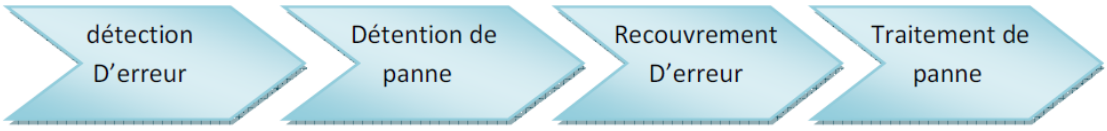

Figure II-2: Procédure générale de tolérance aux pannes

- **Détection d'erreur :** C'est la première phase dans chaque schéma de tolérance aux pannes, dans laquelle on reconnaît qu'un événement inattendu s'est produit. Les techniques de détection de pannes sont généralement classifiées en deux catégories : en ligne et autonome (offline). La détection offline est souvent réalisée à l'aide de programmes de diagnostic qui

s'exécutent quand le système est inactif alors que la détection en ligne vise l'identification de pannes en temps réel et elle est effectuée simultanément avec l'activité du système.

- **Détention de la panne :** Cette phase établit des limites des effets de la panne sur une zone particulière afin d'empêcher la contamination des autres régions. En cas de détection d'intrusion, par exemple, l'isolation des composants compromis minimise le risque d'attaque des composants encore fonctionnels.
- **Recouvrement d'erreur :** C'est la phase dans laquelle on effectue des opérations d'élimination des effets de pannes. Les deux techniques les plus utilisées sont « masquage de panne » et « répétition »
  - **Masquage de panne :** utilise l'information redondante correcte pour éliminer l'impact de l'information erronée.
  - **Répétition :** après que la panne soit détectée, on effectue un nouvel essai pour exécuter une partie du programme, dans l'espoir que la panne soit transitoire.
- **Traitement de panne :** Dans cette phase, la réparation du composant en panne isolé est effectuée. La procédure de réparation dépend du type de la panne. Les pannes permanentes exigent une substitution du composant avec un autre composant fonctionnel. Le système doit contenir un ensemble d'éléments redondants (ou en état standby) qui servent à remplacer les capteurs défaillants.

II.3.3 Fonctionnement

La propriété de tolérance aux pannes est définie par l'habilité du réseau à maintenir ses fonctionnalités sans interruptions provoquées par la panne des capteurs. Elle vise donc à minimiser l'influence de ces pannes sur la tâche globale du réseau. Cette propriété *R(t)* est modélisée dans [13] par une distribution de poisson où *R(t)* donne la probabilité de ne pas avoir une panne pour un nœud capteur pendant l'intervalle de temps [0,t].

$$R(t) = exp\ (-\lambda k\ t)$$

Où λ est le taux de pannes du nœud capteur *k*, et *t* est la période de temps.

Les protocoles conçus pour les réseaux de capteurs doivent atteindre le niveau de tolérance aux pannes requit par le réseau, cela dépend essentiellement de l'environnement de déploiement du réseau, des caractéristiques des micro-capteurs, etc. En effet, si le réseau de capteurs est destiné aux environnements avec un faible degré d'interférences, tel que ceux utilisés dans les bâtiments pour surveiller le taux d'humidité et le degré de température, les protocoles utilisés ne doivent pas cibler une grande tolérances aux pannes, car dans ce type de réseaux, il n'existe pas une grande interférence avec l'environnement, et ses nœuds ne sont pas exposés au risque d'endommagement. Par contre, si le réseau est destiné aux applications militaires telle que la surveillance et le contrôle d'un champ de bataille, le niveau de tolérance aux pannes visé par les protocoles employés doit être très élevé, car les nœuds sont exposés à un grand risque

d'endommagement par des actions hostiles, et les informations captées sont très critiques. Par conséquent, le niveau de tolérance aux pannes requis dépend de l'application du réseau de capteurs conçu, et les schémas de conception doivent prendre en charge ce paramètre.

II.4 Classification des solutions de tolérance aux pannes

Les solutions et les approches de tolérance aux pannes peuvent être vues de plusieurs angles différents. De ce fait, un ensemble de critères est défini pour les classifier. Des catégories de trois classifications distinctes peuvent être citées :

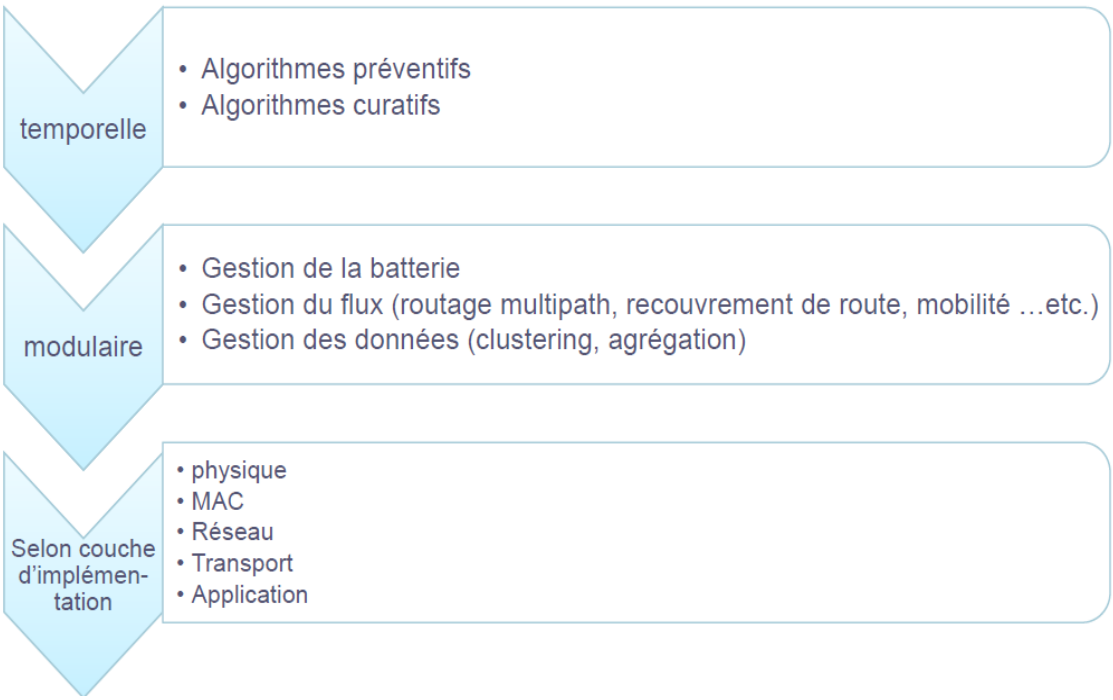

Figure II-3: Classification des solutions de tolérance aux pannes

II.4.1 Classification temporelle

Dans la classification temporelle, nous divisons l'ensemble des algorithmes en deux catégories, et cela selon la phase de traitement. Si le traitement est effectué avant la panne, on parle donc d'algorithmes préventifs sinon les algorithmes sont dits curatifs.

- ***Algorithme préventif :*** implémente des techniques tolérantes aux pannes qui tentent de retarder ou éviter tout type d'erreur afin de garder le réseau fonctionnel le plus longtemps possible. La conservation d'énergie à titre d'exemple, permet de consommer moins d'énergie et évite donc une extinction prématurée de la batterie ce qui augmente la durée de vie des nœuds.
- ***Algorithme curatif :*** utilise une approche optimiste, où le mécanisme de tolérance aux pannes implémenté n'est exécuté qu'après la détection de pannes. Pour cela, plusieurs algorithmes de recouvrement après l'occurrence des pannes sont proposés dans la littérature, par exemple: le recouvrement du chemin de routage, l'élection d'un nouvel agrégateur, etc.

II.4.2 Classification architecturale

Cette classification traite les différents types de gestion des composants, soit au niveau du capteur individuellement ou bien sur tout le réseau. Nous distinguons trois catégories principales:

- **Gestion de la batterie** : Cette catégorie est considérée comme une approche préventive, où les protocoles définissent une distribution uniforme pour la dissipation d'énergie entre les différents nœuds capteurs ; afin de mieux gérer la consommation d'énergie et augmenter ainsi la durée de vie de tout le réseau. En outre, le mécanisme de mise en veille est une technique de gestion de batterie. En effet, les protocoles déterminent des délais de mise en veille des nœuds capteurs inactifs pour une meilleure conservation d'énergie. [14]
- **Gestion de flux** : Cette catégorie regroupe les techniques qui définissent des protocoles de gestion de transfert des données (routage, sélection de canal de transmission, etc.). Nous pouvons trouver des approches préventives ou curatives sur les différentes couches (réseau, liaison de données, etc.) telles que : [14]
  - **Routage multipath** : utilise un algorithme préventif pour déterminer plusieurs chemins depuis chaque capteur vers le nœud collecteur. Ceci garantit la présence de plus d'un chemin fiable pour la transmission et offre une reprise rapide du transfert encas de panne sur le chemin principal et choisissant un des chemins qui restent.
  - **Recouvrement de routes**: après la détection de panne, une technique curative permet de créer un nouveau chemin qui soit le plus fiable pour retransmettre les données.
  - **Allocation de canal**: cette solution est implémentée au niveau de la couche MAC. Elle permet d'effectuer une allocation du canal de transmission d'une manière à diminuer les interférences entre les nœuds voisins et éviter les collisions durant le transfert.
  - **Mobilité**: certains protocoles proposent comme solution tolérante aux pannes la sélection d'un ensemble de nœuds mobiles chargés de se déplacer entre les capteurs et collecter les données captées. Ceci réduira l'énergie consommée au niveau de chaque capteur en éliminant sa tâche de transmission. Un nœud mobile est généralement doté d'une batterie plus importante que celle d'un nœud capteur.
- **Gestion des données :** Les protocoles classés dans cette catégorie offrent une meilleure gestion de données et de leur traitement. Deux principales sous-catégories sont déterminées [16] :
  - **Agrégation**: cette approche est considérée comme une approche préventive. Elle permet d'effectuer un traitement supplémentaire sur les données brutes captées depuis l'environnement. Un nœud agrégateur combine les données provenant de plusieurs nœuds en une information significative. Ce qui réduit considérablement la quantité de données transmises en consommant moins d'énergie pour leur dissémination. Ceci permet donc d'augmenter la durée de vie du réseau. En outre, elle aussi d'amortir l'erreur si le résultat de l'agrégation est une moyenne.

- **Clustering** : une des importantes approches pour traiter la structure d'un réseau de capteurs est le clustering. Il permet la formation d'un backbone virtuel qui améliore l'utilisation des ressources rares telles que la bande passante et l'énergie. Par ailleurs, le clustering aide à réaliser du multiplexage entre différents clusters. En outre, il améliore les performances des algorithmes de routage. Plusieurs protocoles utilisent cette approche préventive et parfois elle est considérée comme une approche curative.

II.5 Approches tolérantes aux pannes dans les RCSF

Dans cette section, on présente certaines approches tolérantes aux pannes en particulier les approches tolérantes aux pannes dans la couche réseau.

Les réseaux de capteurs utilisent une communication multi-sauts. Comparée à une communication sans fil à longue distante, la communication multi-sauts présente une bonne solution pour le problème de propagation de signal et effets de dégradation de signal. De plus, il est recommandé pour ce type de communication de choisir le meilleur chemin qui soit de lien fiable, consomme le moins d'énergie et assure la livraison des données au collecteur. Les solutions proposées à ce niveau de couche sont classifiées en trois principales catégories : routage, agrégation et clustering.

II.5.1 Solutions de routage tolérantes aux pannes

Les protocoles de routage permettent de choisir les meilleurs chemins pour acheminer la donnée depuis la source vers l'utilisateur final. Par ailleurs, ils permettent de sélectionner un chemin de secours en cas d'échec d'envoi sur la route principale ; à cause d'une panne au niveau d'un ou plusieurs capteurs de cette route [16].

a) Algorithme PEQ (Periodic, Event-driven, Query-based)

La motivation de cet algorithme vient du besoin de fournir un support pour toutes les contraintes : faible latence, fiabilité, recouvrement rapide en cas de panne et conservation d'énergie. PEQ combine la conservation d'énergie avec le routage multi-chemins en sélectionnant parmi toutes les routes disponibles, celles qui consomment moins d'énergie. En plus de ce mécanisme préventif qui permet un routage fiable, un mécanisme de recouvrement de pannes est implémenté. Ce dernier remplace le chemin en panne par une autre route qui soit de liens fiables et consomme moins d'énergie. Ainsi, le protocole PEQ couvre la procédure de tolérance aux pannes par la gestion de la consommation d'énergie, la sélection des meilleures routes puis leur recouvrement en cas de panne. [17]

PEQ introduit le paradigme Publish/Subscribe comme montre la figure II.4 [17] pour l'interaction entre le collecteur et les capteurs simples. En effet, les capteurs envoient des notifications d'événements au collecteur, qui va souscrire son intérêt pour certaines de ces informations. Les capteurs concernés publient par la suite l'information désirée.

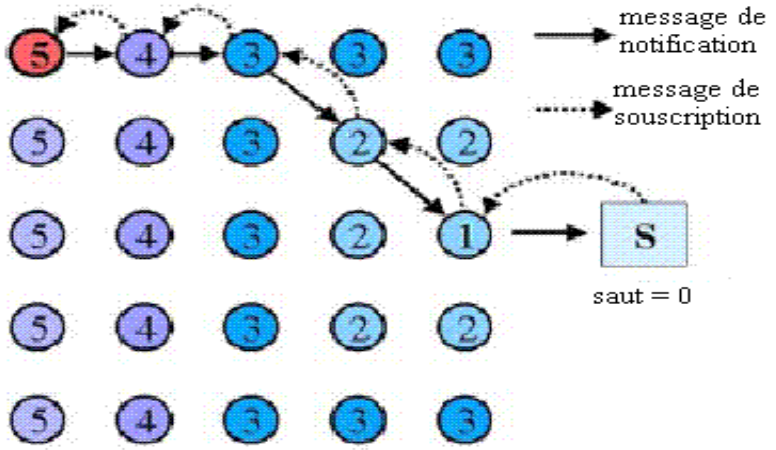

Figure II-4: Mécanisme Publish/Subscribe

Les quatre principales phases du protocole sont:

- **Construction de l'arbre de routage :** cet arbre permet de définir les différents chemins multi-sauts possibles pour acheminer les données de chaque nœud à la station de base. Le collecteur commence le processus en initialisant la variable « saut » à 0; par la suite, chaque capteur prend la valeur du saut actuelle, l'incrémente puis l'envoie à tous ses voisins. Ainsi la valeur au niveau de chaque capteur désigne le nombre nécessaire de sauts pour communiquer avec le collecteur. A la fin de cette phase seulement les meilleurs chemins sont enregistrés.

- **Transmission de paquets de notification** : chaque capteur envoie selon sa table de routage et l'événement capté, une notification de l'information qu'il a à sa disposition. Pour cela, il utilise le chemin le plus rapide et le moins coûteux en terme d'énergie.

- **Propagation des paquets de souscription** : dans cette étape, après une souscription, par le collecteur, des données à transmettre, chaque capteur achemine cette dernière jusqu'au capteur concerné.

- **Mécanisme de recouvrement de route :** le recouvrement est effectué après détection de pannes (figure II.5) [17]. Un capteur envoie son paquet puis attend un acquittement ACK. S'il le reçoit, le message a été bien transmis ; sinon une panne est détectée au niveau du chemin de routage. On effectue donc une recherche "SEARCH" pour la sélection d'un autre capteur destination tout en minimisant le coût du nouveau chemin. Si aucun capteur n'est trouvé (tous les voisins sont détruits) le capteur devient isolé et doit donc augmenter son rayon de transmission radio pour atteindre les capteurs voisins lointains.

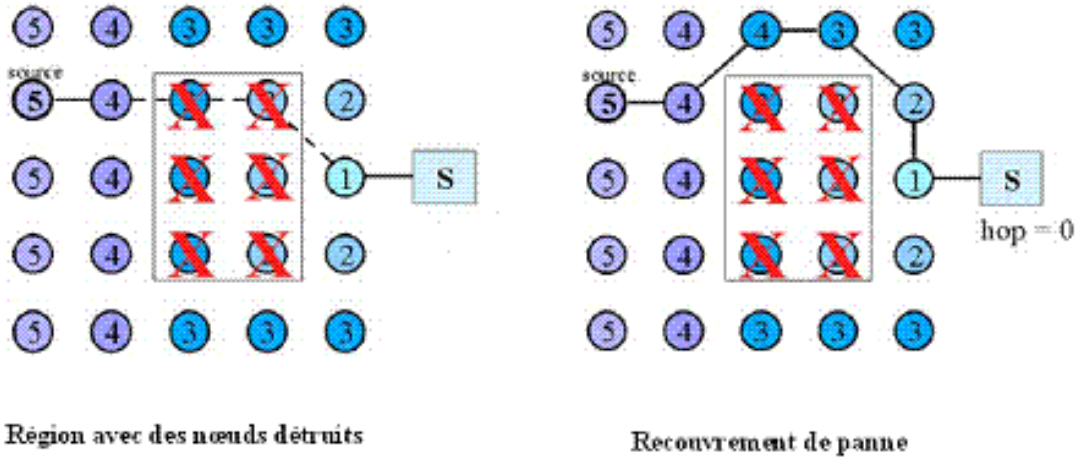

Figure II-5: Recouvrement de routes dans PEQ

b) Protocole EAR

Une solution hybride pour la tolérance aux pannes est proposée dans le protocole EAR qui a un aspect préventif. EAR offre une meilleure conservation d'énergie et définit plusieurs chemins de routage afin de garantir une fiabilité du transport et d'augmenter la durée de vie du réseau. En outre, un mécanisme de recouvrement de pannes est implémenté. Le protocole EAR supporte des réseaux de capteurs à collecteurs multiples (plusieurs nœuds puits). Chaque capteur génère un paquet RPT contenant des informations pour les préférences de l'utilisateur. Les paquets RPT peuvent être envoyés vers n'importe quel collecteur. Cependant, pour chaque capteur intermédiaire le protocole de routage choisit le meilleur chemin qui réduit la consommation d'énergie et la latence. EAR s'exécute selon les étapes suivantes : [10]

- **Phase d'initialisation :** Cette phase permet la construction de l'arbre de routage contenant tous les chemins possibles pour la dissémination des données. Chaque collecteur diffuse un message d'avertissement ADV demandant des paquets RPT. Seuls les capteurs voisins du collecteur qui reçoivent le message ADV, enregistrent le chemin dans leur table de routage ; sans qu'ils propagent le message ADV vers les autres capteurs, comme le montrent les étapes a) et b) de la figure II.6 [10] Les autres capteurs envoient une demande RREQ (Route Request) cherchant un chemin vers le collecteur (étape c). Si un capteur ayant déjà une route stockée dans sa table, reçoit RREQ, il envoie un paquet RREP (Route Reply) à son capteur voisin concerné par la demande (étapes d ; e). Le processus d'initialisation se termine quand chaque capteur reçoit une réponse RREP suite à sa requête RREQ ; puis enregistre le chemin dans sa table de routage.

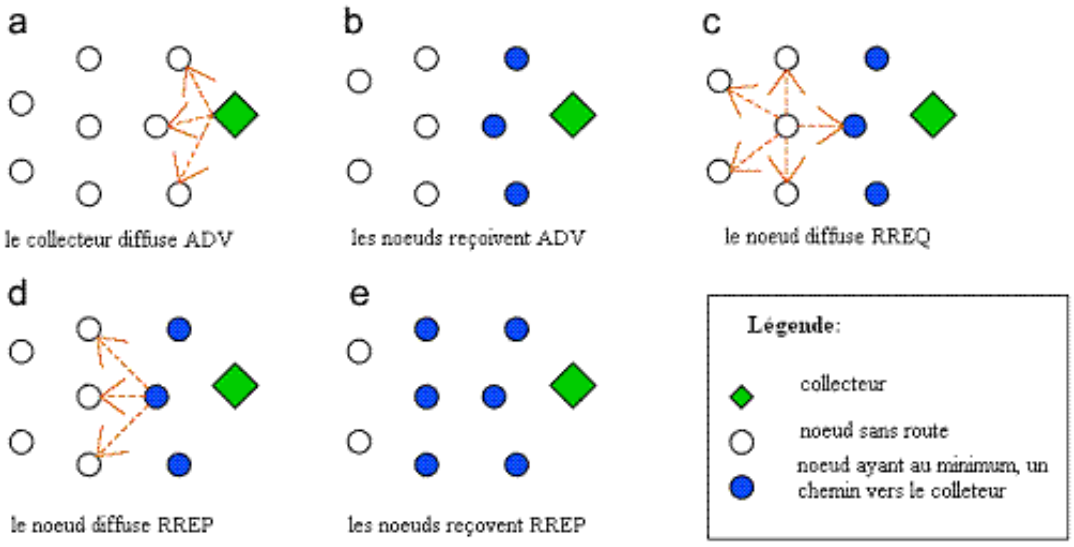

Figure II-6: Fonctionnement du protocole EAR

- **Phase de gestion de route :** Les micro-capteurs, avec leur mémoire de taille réduite, ne peuvent garder tous les chemins possibles dans leurs tables de routage. Pour cela, et afin d'assurer une bonne tolérance aux pannes, on doit garder que les meilleurs chemins. Le protocole EAR définit donc deux métriques pour la sélection des meilleures routes à mémoriser. La première métrique est le nombre de sauts dans une route. Ceci permet de choisir le chemin le plus court. Cependant, la qualité des liens RF n'est pas prise en considération ; dans ce cas, le plus court chemin n'assure pas forcément la fiabilité de transmission. En effet, si un chemin échoue à transmettre N paquets consécutifs, il sera mis dans une « liste noire » l'écartant ainsi d'une future utilisation. La deuxième métrique, appelée Score de route, est définie comme suit :

$$RS = PE \times WE + PT \times WT$$

  - PE: niveau de l'énergie du capteur du prochain saut ;
  - WE: poids assigné à PE dans l'intervalle [0-1] ;
  - PT: taux de succès dans la transmission ;
  - WT: poids assigné à PT dans [0-1] tel que (WT + WE = 1);

  L'historique enregistré sur l'état du routage dans tout le réseau permet, par conséquence, d'adapter la sélection des chemins en choisissant toujours les meilleurs chemins en termes de fiabilité de liens et de conservation d'énergie. Ceci garantit une bonne tolérance aux pannes en évitant la sélection des mauvaises routes.

- **Phase dissémination de données :** Après les deux premières étapes, chaque capteur aura au moins un chemin vers le collecteur. Les capteurs commencent donc à générer des paquets RPT, et le routage des données utilise la métrique « score de route » pour définir le meilleur chemin à emprunter. En cas où ce dernier présente une panne au niveau d'un ou plusieurs de ses capteurs, un mécanisme de recouvrement de route est exécuté, afin d'élire un second chemin fiable pour transmettre les données depuis le capteur vers le collecteur. Par ailleurs, au moment de sa durée d'inactivité, chaque capteur est mis en veille afin d'épargner davantage son énergie et augmenter ainsi la durée de vie de tout le réseau.

c) VTRP (Variable Transmission Range Protocol)

VTRP est une solution d'ajustement du rayon de transmission pour une meilleure propagation de données. Il permet de remédier au problème d'obstacles en les évitant par l'augmentation du rayon de transmission. Ce dernier augmente la probabilité d'atteindre des capteurs actifs quand le rayon actuel utilisé ne couvre aucun capteur à cause de pannes ou d'inactivité des capteurs voisins ou encore dans le cas des réseaux à faible densité. En outre, VTRP offre une meilleure longévité du réseau en évitant l'utilisation fréquente des capteurs critiques (les voisins proches du collecteur) ceci permet d'alléger leur fonction de routage ; conserve leurs batteries et augmente ainsi la durée de vie de tout le réseau. [18]

VTRP utilise des rayons de transmissions variés pour la propagation de données i.e. il permet d'augmenter les rayons de transmissions de différentes manières. Soit k capteurs avec k informations. Le problème posé dans cette étude est donc « comment acheminer toutes les k informations au collecteur d'une manière fiable et efficace ? ». VTRP s'exécute comme suit :

- **Phase de recherche :** Soit p' un capteur qui a reçu une information E de p. Dans la phase de recherche, p' utilise une diffusion périodique de message afin de découvrir le nœud p le plus proche du collecteur (meilleur chemin de p' vers le collecteur en passant par p). Cependant, une détection de panne est possible si aucun nœud p n'est trouvé. Trois différentes raisons sont possibles pour un tel échec : soit le nœud p est mis en veille et ne peut donc répondre à p'; soit il est en panne (détruit, batterie épuisée...etc.) ou bien à cause d'un obstacle qui empêche la communication entre les deux capteurs.

- **Phase de transmission directe :** En cas où la phase de recherche se termine avec succès, p' envoie l'information à p et envoie un message « succès » à p.

- **Phase d'ajustement du rayon de transmission :** Si la phase de recherche échoue (aucun nœud p n'est détecté) p' passe à la phase d'ajustement de son rayon de transmission qui représente le cas de recouvrement après pannes. En effet, chaque capteur maintient un compteur local initialisé à 0. A chaque échec de l'étape de recherche, le compteur est incrémenté, et le rayon de transmission R est modifié. Quatre différentes fonctions sont définies selon la vitesse de variation du rayon de transmission : linéaire, multiplicative, exponentielle et aléatoire :

- **Progrès constant :** VTRP est convenable dans ce cas des réseaux où un grand nombre de capteurs est compromis ;
- **Progrès multiplicatif :** VTRPm définit un rayon de transmission qui est augmenté d'une manière radicale. Ce changement offre une meilleure probabilité pour trouver des capteurs actifs. En revanche, il requiert une consommation d'énergie plus importante.
- **Progrès exponentiel** *:* VTRPp est une variante qui augmente le rayon d'une vitesse encore plus rapide ;
- **Progrès aléatoire :** quand la densité du réseau n'est pas connue au préalable, on utilise l'approche aléatoire VTRPr pour éviter un mauvais comportement du réseau suite à un mauvais choix.

II.5.2 Solutions basées sur le clustering pour la tolérance aux pannes

Les algorithmes proposés dans cette catégorie permettent d'améliorer les performances du processus d'auto-organisation du réseau. Les protocoles du clustering divisent le réseau en un ensemble de clusters ayant chacun un clusterhead qui récupère les données depuis tous les capteurs de son cluster puis les achemine vers le collecteur. Cette solution permet de mieux gérer le trafic de réseau et d'alléger la quantité d'informations qui circule, en effectuant des traitements au sein du cluster avant de propager les données vers le reste du réseau pour les transmettre au collecteur. [18]

a) Protocole CPEQ

En plus de tous les mécanismes de tolérance aux pannes qu'implémente PEQ, la variante CPEQ (Cluster-based PEQ) ajoute un module de clustering pour offrir une meilleure gestion de routage. En effet, les capteurs ayant le plus d'énergie résiduelle sont sélectionnés comme des nœuds agrégateurs (appelés aussi clusterheads). Un nœud agrégateur établit son cluster, et les capteurs appartenant à ce dernier envoient leurs données à l'agrégateur qui effectue d'éventuel traitement sur les données brutes puis les achemine vers le collecteur. Chaque capteur du réseau peut devenir agrégateur pendant une certaine période de temps selon son niveau de batterie. Le but principal de CPEQ est de distribuer d'une manière uniforme la dissipation d'énergie entre les capteurs, et de réduire la latence et le trafic de données dans le réseau. Le protocole CPEQ s'exécute en cinq étapes : [17]

- **Configuration initiale :** Cette phase est exécutée de la même manière que dans l'algorithme PEQ ; où chaque capteur commence par un mécanisme de diffusion pour configurer tout le réseau et connaître par la suite le nombre de sauts nécessaires pour atteindre le collecteur le plus proche. En outre, CPEQ introduit un champ additionnel contenant le pourcentage des capteurs qui deviendront agrégateurs.
- **Sélection d'agrégateur :** C'est la phase d'élection des clusterheads. Après la configuration initiale, chaque capteur peut devenir agrégateur avec un pourcentage donné. En effet, chaque capteur génère un nombre aléatoire entre 0 et 1. Si ce nombre est inférieur à une probabilité p

(probabilité pour devenir agrégateur), le capteur demande à tous ses voisins directs leur niveau de batterie en envoyant un paquet REQ_EN (Request Energy). Chaque voisin répond par un message REP_EN (Reply Energy) contenant son ID et la quantité d'énergie. Le capteur choisit le voisin ayant le maximum d'énergie et diffuse un SET_AGR (Set Aggregator) pour informer tous les capteurs du nouvel agrégateur. Les trois étapes de cette phase sont illustrées dans la figure suivante (figure II.2)[17].

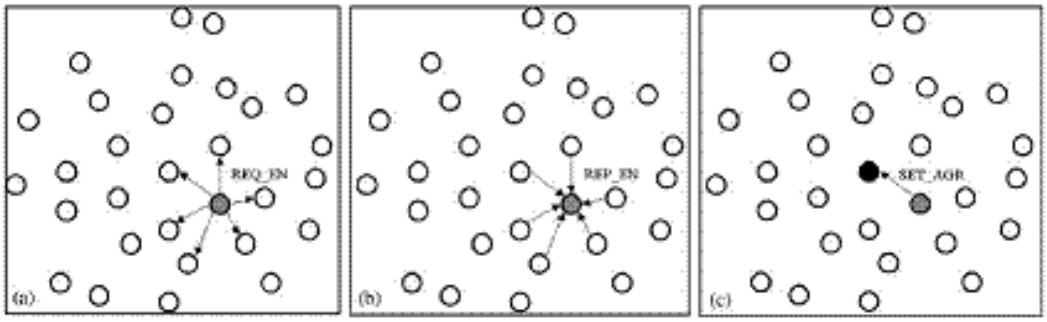

Figure II-7: Configuration initiale

- **Configuration de clusters :** Cette phase divise le réseau en un ensemble de clusters. Le nouveau capteur agrégateur sélectionné doit avertir ses voisins de son rôle d'agrégation. De cette manière chaque agrégateur construit son cluster. La configuration des clusters est réalisée à l'aide des messages AGR_NTF (Aggregator Notification) avec un champ TTL pour limiter la propagation du paquet sur les capteurs se trouvant à une distance inférieure ou égale au TTL. Chaque fois qu'un capteur reçoit ce message, il enregistre l'ID du capteur émetteur dans sa table de routage pour déterminer le chemin vers l'agrégateur. Si un capteur reçoit plusieurs messages AGR_NTF ; il choisit l'agrégateur avec le moindre nombre de sauts. La figure suivante [18] illustre la configuration de clusters avec un TTL=2.

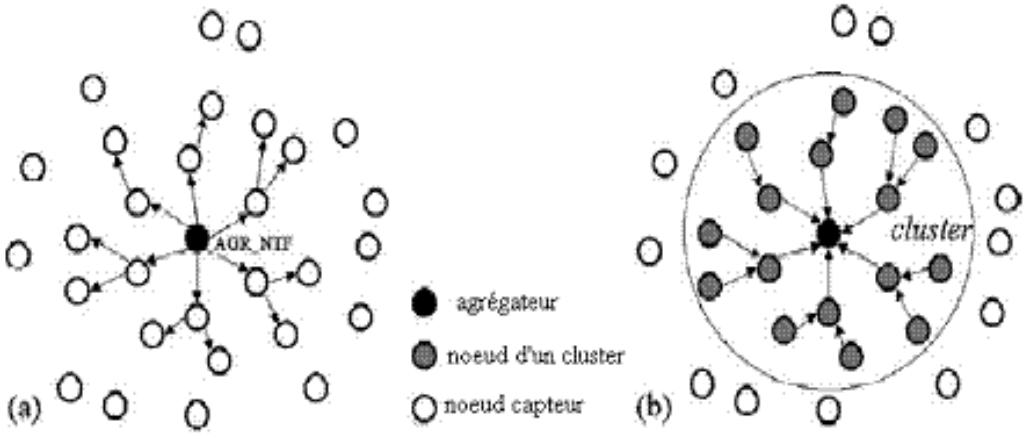

Figure II-8: Configuration des clusters

- **Transmission de données à l'agrégateur :** L'algorithme de routage des données est le même implémenté dans le protocole PEQ. Chaque capteur utilise sa table de routage pour envoyer la donnée vers son agrégateur. Dans CPEQ, l'agrégateur peut être considéré comme un nœud puits. Le mécanisme de recouvrement de chemin est aussi hérité du protocole PEQ.

- **Transmission de données au collecteur** : Après réception des données depuis les capteurs de son cluster, l'agrégateur doit acheminer ces données au collecteur. CPEQ utilise une communication multi-sauts entre l'agrégateur et le collecteur comme montre la figure II.9 [18].

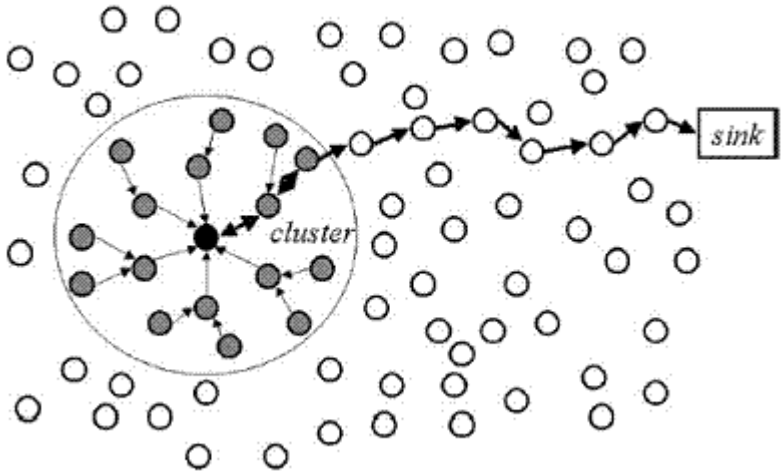

Figure II-9: Transmission des données au collecteur

b) Algorithme K-CDS

On modélise un réseau de capteurs par le graphe G = (V, E) où V est l'ensemble des capteurs et E est l'ensemble des liens sans fil entre ces capteurs.

- **Définition 1 :** un réseau G est k-connexe s'il n'aura aucune partition quand l'on omet i capteur du graphe (i = 1, 2, ..., k-1).i.e. G est k-connexe si chaque deux capteurs du graphe sont connectés par au moins k chemins disjoints.

- **Définition 2 :** un sous-ensemble V' , V est un k-DS (k-dominating Set) de G si chaque nœud de V qui n'appartient pas à V' a au moins k voisins dans V'. Le k-DS V' devient k-CDS si le sous-graphe G[V'] est k-connexe.

L'algorithme K-CDS utilise une approche préventive basée sur le clustering. Il propose une construction d'un ensemble k-connexe dominant k-CDScomme un backbone virtuel pour offrir une efficacité de routage aussi bien qu'une bonne tolérance aux pannes. Pour cela, quatre approches ont été introduites ; dont deux sont des algorithmes probabilistes, une est déterministe

et la dernière est une hybridation des approches déterministes et probabilistes. Ces quatre approches permettent de considérer différents critères pour la construction des clusters.

c) KAT-Mobility

Dans KAT-mobiliy (K-means And TSP-based mobility), en plus du clustering, le concept de mobilité est implémenté au niveau des nœuds collecteurs. Ces deux mécanismes, définissent une technique préventive hybride tolérante aux pannes qui offre une meilleure gestion d'énergie et augmente donc la durée de vie du réseau. Après réorganisation du réseau en clusters, la méthode proposée pilote le collecteur mobile pour se déplacer à travers les centres des clusters en prenant le chemin optimal. Le collecteur mobile récupère donc les données depuis les capteurs des clusters visités.

Le principe de KAT-mobility se résume en deux procédures : clustering et optimisation du routage. La figure II.10 illustre le principe de KAT-mobility. [6]

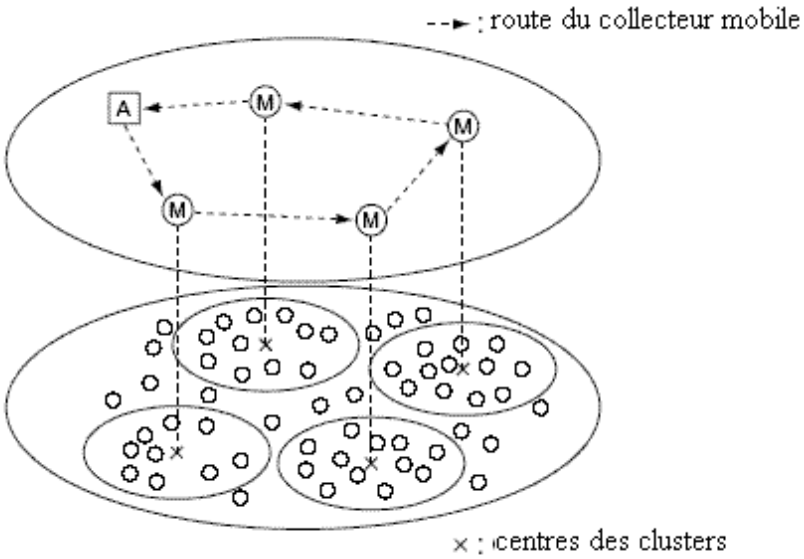

Figure II-10: Illustration de l'algorithme KAT-mobility

- **Algorithme de clustering:** Cette procédure divise l'ensemble des N capteurs en k clusters C_1, C_2, ..., C_k. Le coût du cluster est évalué par l'erreur approximative entre le collecteur et les nœuds capteurs. Soit d(x, yi) cette erreur, où x est un capteur et yi est un collecteur (i = 1, 2, ..., k). d(x, yi) est défini par la distance euclidienne entre le capteur et le collecteur. Le but est donc, d'affecter chaque capteur à un cluster C_i en minimisant l'erreur totale des clusters.
- **Optimisation du routage :** Trouver un chemin optimal pour le nœud mobile est identique au problème du voyageur de commerce (TSP). Ainsi ; un collecteur représente le voyageur, et les centres des clusters définissent les villes. L'optimisation de la route du collecteur mobile pour

visiter tous les centres des clusters une et une seule fois est équivalente à la recherche du plus court voyage d'un commerçant pour visiter chaque ville une seule fois.

Les résultats de simulation ont montré que KAT-mobility peut fournir une meilleure conservation d'énergie aussi bien qu'une bonne tolérance aux pannes en cas de mal fonctionnement de certains capteurs.

II.5.3 Solutions basées sur l'agrégation des données

Minimiser la consommation d'énergie revient à minimiser, entre autre, la quantité de données transmises dans le réseau en particulier les données redondantes. En effet, d'après les statistiques, 70% de l'énergie consommée dans un nœud capteur est due aux transmissions. L'agrégation combine les données provenant de plusieurs capteurs en une information significative ; en éliminant ainsi la redondance. Ceci résout le problème d'implosion dans le routage et allège ainsi la congestion du réseau.

a) Classification des protocoles d'agrégation

On peut classer les différentes techniques d'agrégation de données dans les réseaux de capteurs en deux approches, comme illustré par la figue II.11 [4].

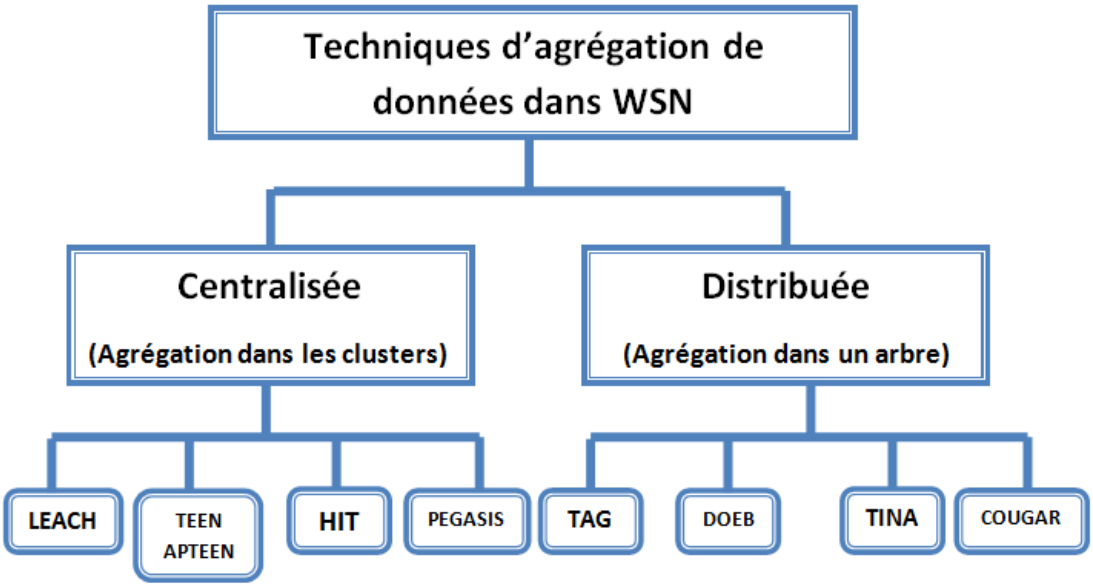

Figure II-11: Protocoles d'agrégation de données dans les RCSF

Parmi ces protocoles on choisit de détailler trois protocoles courants qui sont : LEACH, TEEN et COUGAR:

- **LEACH (Low-Energy Adaptive Clustering Hierarchy) :** LEACH (Low-Energy Adaptive Clustering Hierarchy) est un protocole de routage hiérarchique, employant un procédé de clustering qui divise le réseau en deux niveaux : les cluster-heads et les nœuds membres. Le protocole se déroule en rounds. Chaque round se compose de deux phases : construction et communication. [4]
- **TEEN (Threshold sensitive Energy Efficient sensor Network protocol) :** En utilisant TDMA, le protocole LEACH est destiné aux applications time-driven. Dans ce type d'application, la donnée est propagée d'une manière périodique. Cependant, ce genre de protocole est inadapté pour les applications event-driven, où un comportement réactif est nécessaire pour le bon fonctionnement du système. TEENa été développé pour améliorer LEACH afin de répondre aux exigences des applications event-driven. [19]

  La majorité du comportement de TEEN est semblable au protocole LEACH. Cependant, quelques différences existent. Les chefs élus ne transmettent pas un schedule TDMA, mais émettent un message contenant les informations suivantes :

  - Attributs : représentent la tâche demandée au capteur.
  - Hard threshold (HT) : détermine la valeur critique après laquelle les membres doivent envoyer leur rapports de données.
  - Soft threshold (ST) : spécifie le changement minimal obligeant le nœud à envoyer un nouveau rapport.

  Donc, lorsqu'un nœud s'aperçoit que la valeur captée a dépassé HT, il doit émettre un rapport au chef. Il ne réémet un nouveau rapport que si la valeur change radicalement, i.e. la différence dépasse ST. Ce mécanisme permet d'implémenter un comportement réactif, tout en limitant le nombre de messages utilisés.

- **COUGAR :** Dans Cougar, les données produites par le réseau de capteurs sont modélisées comme une table relationnelle. Dans cette table, chacun des attributs représente soit des informations sur le capteur ou bien des données produites par ce capteur. L'approche Cougar fournit une agrégation partielle au niveau des capteurs. Chaque capteur maintient une liste d'attente contenant les capteurs fils qui doivent lui envoyer les paquets. Le capteur n'émet le paquet agrégé au prochain saut que s'il a reçu les paquets de tous les capteurs de la liste d'attente. Cependant, un capteur peut devenir inaccessible à cause du mouvement ou d'un problème de batterie. Pour cela, Cougar utilise un Timer afin d'éviter une attente indéfinie [4].

II.6 Conclusion

Dans ce chapitre, on a présenté la tolérance aux pannes dans les réseaux de capteurs et les différentes approches proposées dans cette optique. Puis, on fait une classification sur la tolérance aux pannes.

Dans le chapitre qui suit, on présente notre contribution qui consiste à instaurer un mécanisme de tolérance aux pannes dans les réseaux de capteurs. Ce mécanisme a un aspect préventif guidé par les données collectées.

Chapitre 3

Implémentation d’une application de tolérance aux pannes dans les RCSF

Chapitre 3 Implémentation d'une application de tolérance aux pannes

III.1 Introduction

Les réseaux de capteurs ont envahi plusieurs domaines d'applications. On distingue deux types d'applications : les applications orientées événements et les applications orientées surveillance. Dans le premier type, on s'intéresse principalement aux événements pertinents qui peuvent survenir dans la zone de déploiement alors que dans le deuxième type on s'intéresse à une surveillance de longue durée d'une zone d'intérêt. Les applications orientées surveillance consistent à envoyer des informations périodiques à un centre de contrôle distant. Ces informations peuvent être exploitées directement ou stocker dans une base de données pour effectuer des éventuelles statistiques. Par exemple, le suivi d'un patient, le suivi d'un animal, contrôle d'un bâtiment, etc.

Dans ce chapitre, on implémente une application orientée tolérance aux fautes en tenant compte des contraintes matérielles des réseaux de capteurs : puissance de calcul limitée, mémoire de stockage réduite, portée de transmission courte, et une autonomie de l'énergie. Pour se faire, des outils logiciels spéciaux ont été développés pour les dispositifs à ressources limitées comme les réseaux de capteurs.

L'application développée est une application générique. Elle consiste à faire des prélèvements de températures dans plusieurs endroits d'une zone d'intérêt et les envoyer à un poste de contrôle distant qu'il les analyse et les stocke dans des fichiers pour effectuer des éventuelles statistiques qui serviront à détecter des capteurs qui peuvent cesser de fonctionner dans le futur i.e. il s'agit d'une application préventive de pannes.

III.2 Objectifs de l'application

Dans les applications sensibles qui nécessitent une surveillance continue, il est nécessaire d'instaurer un mécanisme préventif de pannes pour éviter toute conséquence catastrophique. De ce fait, il faudrait déployer un grand nombre de capteurs pour garantir l'envoi de la bonne information au centre de contrôle. D'où, quand on remarque qu'un capteur commence à perdre sa fiabilité on devrait ne pas utiliser ses données collectées et le remplacer par un autre. Dans ce contexte, on développe une application

qui permet une détection préventive de pannes à partir de l'analyse des données collectées pendant des périodes différentes.

III.3 Les choix techniques

L'implémentation de cette application nécessite des outils matériels et des outils logiciels bien spécifiques tels que NesC et Java comme langages, TinyOs comme système d'exploitation.

III.3.1 Choix du langage

Pour réaliser l'application, on a utilisé deux langages différents : NesC et Java. Le premier pour programmer les capteurs et le deuxième permet d'utiliser ses atouts pour bien exploiter l'application au niveau du poste de contrôle par exemple la création d'interfaces graphique grâce à sa librairie Swing et aussi la communication dans un réseau grâce au Socket. Ce langage est supporté par Windows, Unix et Mac et dispose d'outils pour communiquer et exploiter les données envoyées par les capteurs tels SerialForwarder et MIG.

Ces deux langages se valant plus ou moins, le choix du langage s'est donc fait par affinité. Java contenant les librairies nécessaires au développement d'interfaces homme/machine et étant le seul dont le code exécutable est portable, c'est ce langage qui a été choisi pour l'application. En outre, NesC est un langage orienté composant conçu pour programmer les dispositifs à ressources limitées.

III.3.2 Système d'exploitation : TinyOS

Le système choisi pour le développement de cette application est un système léger qui est conçu pour les équipements à ressources limitées.

TinyOS reste néanmoins le plus répandu pour les RCSFs car il répond aux exigences particulières des applications des RCSF. Il convient alors de mentionner les propriétés qui rendent TinyOS aussi populaire et réputé pour ce genre de réseaux [17].

- Une taille de mémoire réduite.
- Une basse consommation d'énergie.
- Des opérations robustes.
- Applications orientées composants : TinyOS fournit une réserve de composants systèmes utilisables au besoin.
- Programmation orientée évènement : Généralement sur TinyOS, un programme s'exécute suivant le déclenchement des événements. Sinon, les capteurs restent en veille ce qui maximise la durée de vie du réseau.

III.3.3 Installation logicielle

La première étape a été de prendre en main les capteurs, le langage NesC et le système d'exploitation TinyOs. On a choisi l'environnement Linux (Ubuntu) pour la mise en place de l'application de monitoring. Il est possible de l'exploiter sous cygwin (Windows) mais il y a des soucis avec la deuxième version de TinyOs (TinyOs-2.x) sous cet environnement.

La phase d'installation a été la plus délicate. En effet, on a commencé par installer TinyOS 2.1.1, NesC et le micro-contrôleur MSP430 pour les capteurs de type TelosB. La procédure d'installation de TinyOs se déroule en plusieurs étapes et elle est décrite dans l'annexe.

III.3.4 Installation matérielle

Une fois l'installation logicielle est terminée, il a fallu installer le matériel : une station de base reliée à l'ordinateur via un câble USB, différents capteurs TelosB (Senders). Chacun des Senders mesure la température et la communique à la station de base via une liaison sans fil. Un capteur Sender communique avec la station de base via une liaison sans fil, et la station de base communique avec l'ordinateur via le câble USB. La figure III.1 présente les schémas de communication dans la plateforme qu'on a développée.

Dans cette architecture, on a plusieurs capteurs de type Sender déployés dans une zone. Chacun de ces capteurs communique directement avec la station de base. Il s'agit d'une architecture plate.

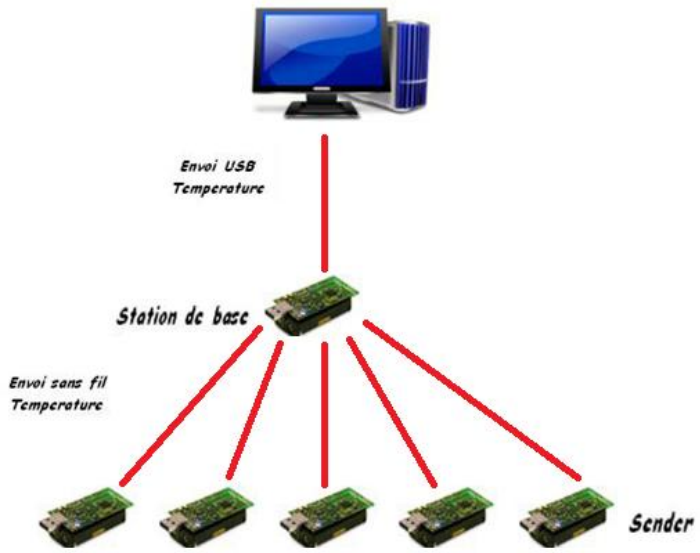

Figure III-1: Architecture de l'application

Pour réaliser l'application on dispose de six capteurs de type TelosB comme montre la figure III.2 :

- Cinq capteurs sont déployés dans une zone d'intérêt et communiquent directement avec la station de base
- Un capteur est configuré comme station de base et il est connecté au poste de contrôle.

Figure III-2: Environnement du travail

Dans cette architecture simplifiée, chaque capteur TelosB Sender envoie périodiquement(durant 5 minutes) la température ambiante à la station de base via une liaison sans fil. La station de base transmet ces températures via la liaison USB à l'ordinateur de contrôle. Ce dernier analyse ces données pour décider sur l'éventuel capteur qui pourrait être défaillant dans le futur.

III.4 Etapes de réalisation de la plateforme

Dans cette section, on détaille les étapes de réalisation de l'application.

III.4.1 Programmes en NesC

Les programmes en NesC qu'on les met sur les capteurs contiennent les composants suivants:

- Main : cœur de l'application.
- ActiveMessage : permet l'accès à la liaison sans fil et l'encapsulation de messages qui pourront être ensuite envoyés via la liaison sans fil.
- Leds : permet de suivre le déroulement du programme en allumant les Leds du capteur à des moments pertinents.
- TimerMilli : permet le déclenchement d'évènements périodiques.

- SensirionSht11 : lit la température et l'humidité.
- AMSender : permet l'envoi de messages via la liaison sans fil.

Le composant TimerMilli déclenche chaque moment un évènement i.e. ce moment est passé en paramètre. Cet événement demande au composant SensirionSht11 de lire et calculer la température puis d'encapsuler cette valeur à l'aide du composant ActiveMessage et de l'envoyer via le composant AMSender. Ces données sont ensuite reçues par la station de base, et seront par suite stockées, traitées et analysées par le centre de contrôle.

NesC permet de déclarer deux types de fichiers : les modules et les configurations.

- Le fichier configuration est la définition du ou des composants qui seront utilisés par l'application déployée sur le capteur.
- Les modules constituent les briques élémentaires de code et implémentent une ou plusieurs interfaces qui sont des fichiers décrivant les commandes et évènements proposés par le composant qui les implémente.

III.4.2 Mise en place de la plateforme

On a déployé cinq capteurs Sender et un capteur jouant le rôle de Station de base connecté à un ordinateur de contrôle via une liaison USB.

- **Le capteur Sender :** Le Sender collecte les données de températures à partir de l'environnement et les envoie à la station de base via le sans fil.
- **Le capteur Station de base :** Reçoit les données collectées par le Sender et les envoie à l'ordinateur de contrôle via une liaison USB.
- **L'ordinateur de contrôle :** Reçoit les données venant de la station de base et les stocke dans des fichiers pour pouvoir procéder à une analyse préventive des données.

III.4.3 Programmation des capteurs

1) Compilation et installation des cinq capteurs Sender et de la Station de base via les commandes présentées sans l'annexe B.
2) Connecter la Station de base à l'ordinateur de contrôle via la liaison USB.
3) Placer les cinq capteurs dans l'environnement de collecte.
4) Démarrer la collecte de données en lançant le programme sous TinyOs.
5) Stoker les données reçues dans des fichiers pour l'analyse de données.
6) Faire l'analyse de données.

III.4.4 Quelques exécutions

a) Compilation et installation

```
root@amel-Compaq-610:/opt/Benahmed# cd BaseStation/
root@amel-Compaq-610:/opt/Benahmed/BaseStation# ls
BaseStationC.nc  BaseStationP.nc  build  Makefile  README.txt
root@amel-Compaq-610:/opt/Benahmed/BaseStation# make telosb
mkdir -p build/telosb
    compiling BaseStationC to a telosb binary
ncc -o build/telosb/main.exe  -Os -O -mdisable-hwmul -fnesc-separator=__ -Wall -
Wshadow -Wnesc-all -target=telosb -fnesc-cfile=build/telosb/app.c -board= -DDEFI
NED_TOS_AM_GROUP=0x22 -DCC2420_NO_ACKNOWLEDGEMENTS -DCC2420_NO_ADDRESS_RECOGNITI
ON -DTASKLET_IS_TASK -DIDENT_APPNAME=\"BaseStationC\" -DIDENT_USERNAME=\"root\"
-DIDENT_HOSTNAME=\"amel-Compaq-610\" -DIDENT_USERHASH=0xfa44528fL -DIDENT_TIMEST
AMP=0x53577db3L -DIDENT_UIDHASH=0x09aedfcaL  BaseStationC.nc -lm
/opt/tinyos-2.1.1//tos/chips/cc2420/lpl/DummyLplC.nc:39:2: warning: #warning "**
* LOW POWER COMMUNICATIONS DISABLED ***"
    compiled BaseStationC to build/telosb/main.exe
          15616 bytes in ROM
           1773 bytes in RAM
msp430-objcopy --output-target=ihex build/telosb/main.exe build/telosb/main.ihex
    writing TOS image
root@amel-Compaq-610:/opt/Benahmed/BaseStation# ls
BaseStationC.nc  BaseStationP.nc  build  Makefile  README.txt
root@amel-Compaq-610:/opt/Benahmed/BaseStation# ls
BaseStationC.nc  BaseStationP.nc  build  Makefile  README.txt
root@amel-Compaq-610:/opt/Benahmed/BaseStation#
```

Figure III-3: Compilation du programme

```
collect1.txt  conn~        MsgReader.class  SenderC.nc~
collect2.txt  conn.java~   MsgReader.java   Stok.java~
collect3.txt  Makefile     MsgReader.java~  TemperatureMsg.class
collect4.txt  Makefile~    panierReq.java~  TemperatureMsg.java
collect5.txt  Message.h    SenderAppC.nc    TemperatureMsg.java~
root@amel-Compaq-610:/opt/Benahmed/Sender# cd ..
root@amel-Compaq-610:/opt/Benahmed# cd BaseStation/
root@amel-Compaq-610:/opt/Benahmed/BaseStation# ls
BaseStationC.nc  BaseStationP.nc  build  Makefile  README.txt
root@amel-Compaq-610:/opt/Benahmed/BaseStation# make telosb install,0 bsl /dev/t
tyUSB0
mkdir -p build/telosb
    compiling BaseStationC to a telosb binary
ncc -o build/telosb/main.exe  -Os -O -mdisable-hwmul -fnesc-separator=__ -Wall -
Wshadow -Wnesc-all -target=telosb -fnesc-cfile=build/telosb/app.c -board= -DDEFI
NED_TOS_AM_GROUP=0x22 -DCC2420_NO_ACKNOWLEDGEMENTS -DCC2420_NO_ADDRESS_RECOGNITI
ON -DTASKLET_IS_TASK -DIDENT_APPNAME=\"BaseStationC\" -DIDENT_USERNAME=\"root\"
-DIDENT_HOSTNAME=\"amel-Compaq-610\" -DIDENT_USERHASH=0xfa44528fL -DIDENT_TIMEST
AMP=0x53577e3cL -DIDENT_UIDHASH=0x890e466eL  BaseStationC.nc -lm
/opt/tinyos-2.1.1//tos/chips/cc2420/lpl/DummyLplC.nc:39:2: warning: #warning "**
* LOW POWER COMMUNICATIONS DISABLED ***"
    compiled BaseStationC to build/telosb/main.exe
          15616 bytes in ROM
           1773 bytes in RAM
msp430-objcopy --output-target=ihex build/telosb/main.exe build/telosb/main.ihex
    writing TOS image
tos-set-symbols --objcopy msp430-objcopy --objdump msp430-objdump --target ihex
build/telosb/main.ihex build/telosb/main.ihex.out-0 TOS_NODE_ID=0 ActiveMessageA
ddressC__addr=0
    found mote on /dev/ttyUSB0 (using bsl,auto)
    installing telosb binary using bsl
tos-bsl --telosb -c /dev/ttyUSB0 -r -e -I -p build/telosb/main.ihex.out-0
MSP430 Bootstrap Loader Version: 1.39-telos-8
Mass Erase...
Transmit default password ...
Invoking BSL...
Transmit default password ...
Current bootstrap loader version: 1.61 (Device ID: f16c)
Changing baudrate to 38400 ...
Program ...
15648 bytes programmed.
```

Figure III-4: Installation de la station de base

```
root@amel-Compaq-610:/opt/Benahmed/Sender# make telosb install,1 bsl /dev/ttyUSB0
mkdir -p build/telosb
javac -target 1.6 -source 1.6 *.java
Note: MsgReader.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
    compiling SenderAppC to a telosb binary
ncc -o build/telosb/main.exe  -Os -O -mdisable-hwmul -fnesc-separator=__ -Wall -Wsh
 -board= -DDEFINED_TOS_AM_GROUP=0x22 -I/opt/tinyos-2.1.1//tos/lib/T2Hack -DIDENT_AP
E=\"amel-Compaq-610\" -DIDENT_USERHASH=0xfa44528fL -DIDENT_TIMESTAMP=0x53577eb4L -D
/opt/tinyos-2.1.1//tos/chips/cc2420/lpl/DummyLplC.nc:39:2: warning: #warning "*** L
    compiled SenderAppC to build/telosb/main.exe
          15874 bytes in ROM
            363 bytes in RAM
msp430-objcopy --output-target=ihex build/telosb/main.exe build/telosb/main.ihex
    writing TOS image
tos-set-symbols --objcopy msp430-objcopy --objdump msp430-objdump --target ihex bui
 ActiveMessageAddressC__addr=1
    found mote on /dev/ttyUSB0 (using bsl,auto)
    installing telosb binary using bsl
tos-bsl --telosb -c /dev/ttyUSB0 -r -e -I -p build/telosb/main.ihex.out-1
MSP430 Bootstrap Loader Version: 1.39-telos-8
Mass Erase...
Transmit default password ...
Invoking BSL...
Transmit default password ...
Current bootstrap loader version: 1.61 (Device ID: f16c)
Changing baudrate to 38400 ...
Program ...
15906 bytes programmed.
Reset device ...
rm -f build/telosb/main.exe.out-1 build/telosb/main.ihex.out-1
root@amel-Compaq-610:/opt/Benahmed/Sender#
```

Figure III-5: Installation des capteurs Sender

b) Exécution et collecte de données

```
  [nodeid=0x3]

numero de la donnee est : 8
1398242480460: Message <TemperatureMsg>
  [temperature=0x14]
  [nodeid=0x1]

numero de la donnee est : 9
1398242481214: Message <TemperatureMsg>
  [temperature=0x15]
  [nodeid=0x5]

numero de la donnee est : 10
1398242481229: Message <TemperatureMsg>
  [temperature=0x15]
  [nodeid=0x2]

numero de la donnee est : 11
1398242481360: Message <TemperatureMsg>
  [temperature=0x13]
  [nodeid=0x3]

numero de la donnee est : 12
```

Figure III-6: Collecte de données

III.5 Analyse des données et détection des anomalies

Dans notre contexte, on suppose que le réseau déployé assure une k-couverture au niveau de la zone d'intérêt i.e. chaque point de la zone est couvert par au moins k capteurs. Cette démarche permet d'assurer la tolérance aux fautes dans ce réseau puisque l'information envoyée au centre de contrôle est redondante. En outre, la collecte de données se fait d'une manière périodique par tous les capteurs et ces données sont stockées dans un fichier pour être analysées par la suite.

Ainsi si on remarque qu'un capteur retourne une valeur qui est différente de la moyenne d'un certain seuil ce capteur peut être considérée comme défaillant et on le remplace au début de la période qui suit. Cette démarche peut être considérée comme une détection préventive de pannes. Les équations suivantes régissent cette démarche :

$$\bar{x} = \sum_{i=1}^{n} x_i / n$$

Où

- $\bar{x}$ la moyenne
- n représente le nombre de capteurs
- x_i la valeur collectée par le capteur i

$$e_x = \sum_{i=1}^{n} (\bar{x} - x_i)^2$$

On calcule l'écart type pour connaitre la répartition des collectes des différents capteurs et de ne pas prendre en considération les données des capteurs qui sont supérieures ou inférieures d'un certain seuil (e_x) de la moyenne. De ce fait, les données concernées seront celles qui respectent la contrainte suivante :

$$|x_i - \bar{x}| < \sqrt[2]{e_x}$$

On suppose qu'il y a p données qui répondent à cette contrainte. D'où la nouvelle moyenne serait calculée selon le schéma algorithmique suivant :

S = 0 ;

p = 0 ; // le nombre de données qui vérifient la contrainte ()

pour i de 1 à n faire

si $|x_i - \bar{x}| < e_x$ alors début

$p = p + 1$

$S = S + x_i$

fsi

fin pour

$\bar{x} = \frac{S}{p}$

$taux = \frac{p}{n}$

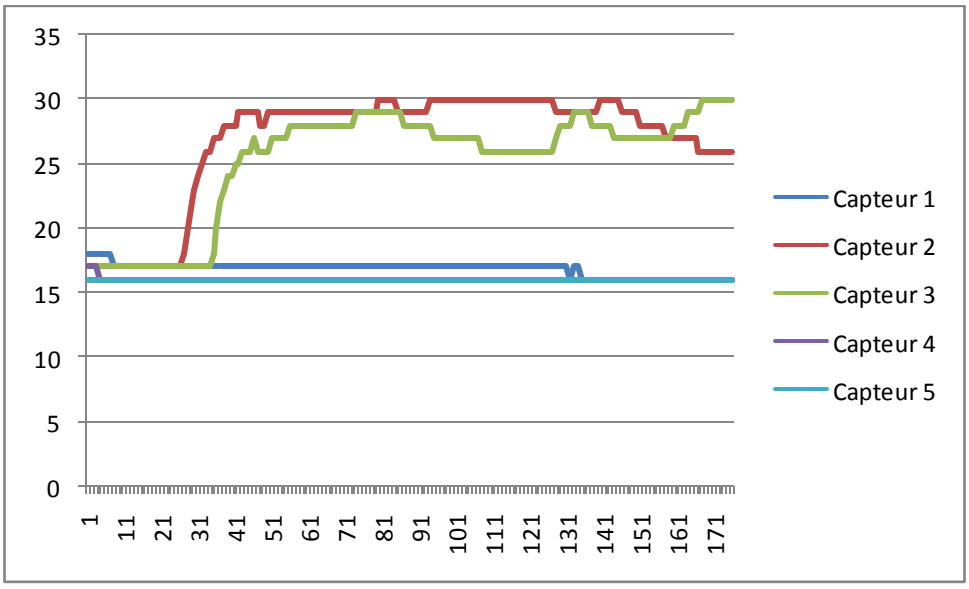

Figure III-7: Variation des données durant la vie du réseau

Les données qu'on a tentées de collecter sont des données sensibles utilisées dans des environnements qui nécessitent une grande précision comme par exemple les environnements industriels, médicaux, militaires, etc...

Pour prévenir la panne avant son occurrence, on a discrétisé le temps de surveillance de l'environnement en plusieurs périodes afin qu'on puisse détecter la panne à une étape précoce. La figure III.8 illustre la discrétisation du temps de surveillance des différents capteurs en des périodes.

On a discrétisé le temps en deux périodes dont lesquelles il y a eu un changement de valeurs assez important et certainement pas négligeable. D'où, dès la première période on pourra décider sur la fiabilité des capteurs.

Première période :

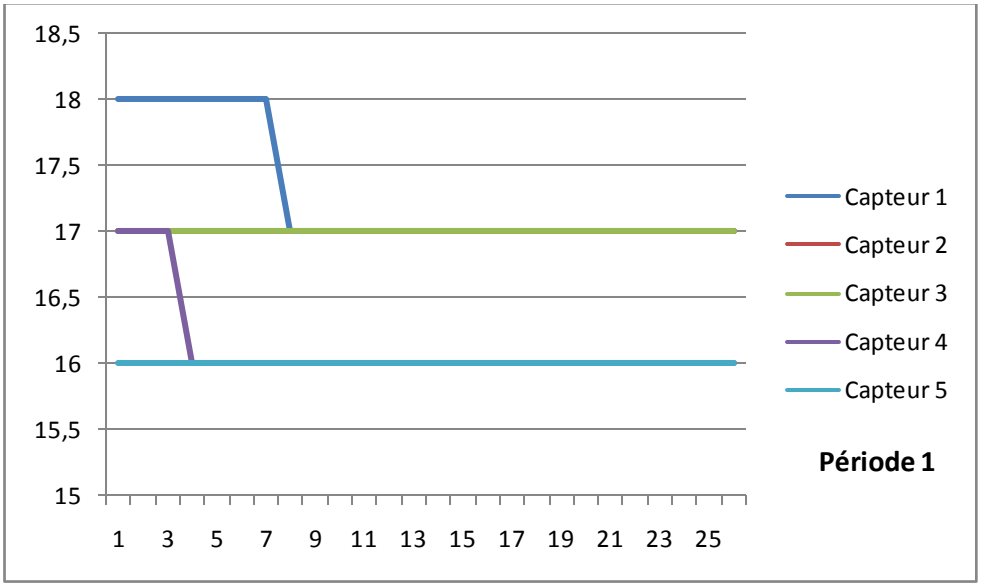

Figure III-8: Variation des données: Période 1

Deuxième période :

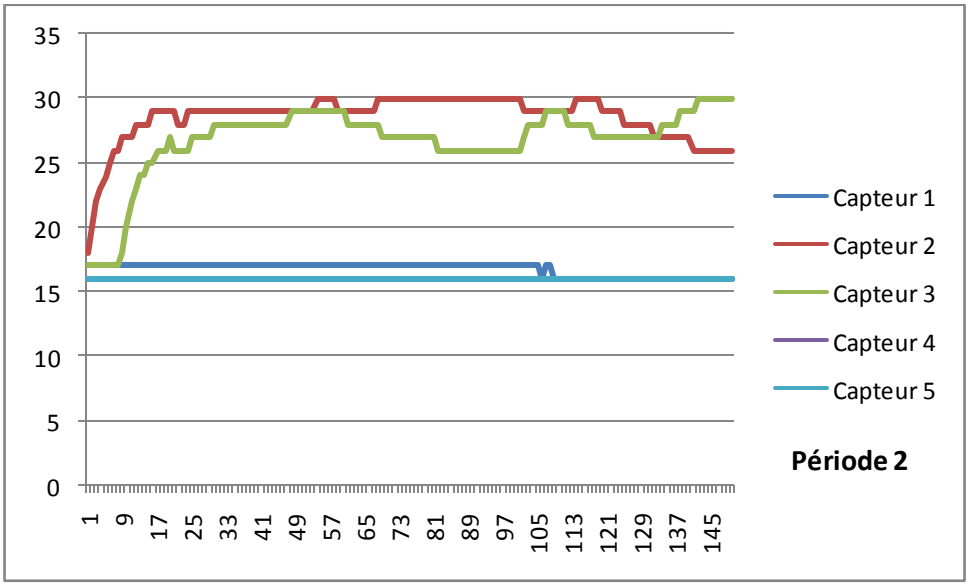

Figure III-9: Variation des données : Période 2

Dans cette deuxième période on a assisté à un changement progressif des données reçues au niveau des deux capteurs : 2 et 3.

III.6 Conclusion

Dans ce chapitre, on a présenté les outils logiciels et matériels ainsi que la démarche à suivre pour réaliser une application permettant la détection d'une panne d'un capteur à une étape précoce.

Cette implémentation basée sur les réseaux de capteurs exige des outils bien particuliers qui sont développés pour exploiter les systèmes à ressources limitées tels que un système d'exploitation léger « TinyOs », un langage orienté composant « NesC» et une base de données pour stocker et évaluer éventuellement des statistiques sur les informations collectées.

Dans cette application, le déploiement des capteurs permet la couverture de chaque point de la zone d'intérêt par plusieurs capteurs. En outre, durant chaque période chaque capteur collecte l'information et l'envoie au centre de contrôle. Puis, une analyse des données collectées est faite pour détecter des éventuelles anomalies. Ainsi, si on remarque qu'il existe des valeurs aberrantes par un des capteurs on considérera ce capteur comme il est défaillant.

Conclusion générale

Conclusion générale

Les réseaux de capteurs sont composés d'un très grand nombre de dispositifs de communication ultra petits, autonomes avec des ressources de calcul et d'énergie limitées. Ils sont actuellement considérés comme l'une des technologies qui bouleverse notre façon de vivre, grâce à leur utilisation dans différents domaines d'application. Cependant, les réseaux de capteurs sans fil rencontrent plusieurs problèmes qui affectent leur bon fonctionnement dû à leurs caractéristiques ; tels que les limitations de batterie, le type de communication, les environnements hostiles où sont déployés les capteurs, etc... Par ailleurs, ces réseaux sont caractérisés par les pannes des nœuds qui peuvent causer un dysfonctionnement du réseau en entier. Dans cette optique, il est commode de proposer des techniques qui permettent de prévoir les pannes à une étape précoce.

Une panne au niveau d'un capteur peut se produire à cause d'une perte de connexions sans fil due à l'extinction du capteur suite à l'épuisement de sa batterie, ou tout simplement à une destruction physique accidentelle ou intentionnelle par un ennemi. Par conséquence, il faut faire face à ces pannes en proposant des techniques tolérantes aux pannes.

Dans ce mémoire, on a réalisé une application qui permet de détecter les pannes dans leurs débuts. Cette détection permet à des applications sensibles d'éviter des catastrophes.

En perspectives, on propose d'implémenter des techniques d'analyse de données quand les données sont nombreuses et variées.

Annexes

Annexe A : Installation de TinyOS 2.1.1 sous Linux

Il y a deux façons de faire une installation propre de TinyOs. La première façon est d'installer une VM qui a une installation complète de TinyOs. La deuxième façon est d'installer TinyOs sur votre système d'exploitation hôte. Lors de l'installation sur un système d'exploitation hôte, vous pouvez soit utiliser un paquet Debian ou installer manuellement avec RPM. On a fait une installation sur un Os avec un paquet Debian.

L'installation se fait en deux étapes sur votre système d'exploitation hôte avec les paquets Debian : [20]

- Retirez tout ancien référentiel TinyOs du fichier : / etc / apt / sources.list et ajoutez la ligne suivante : **deb** http://tinyos.stanford.edu/tinyos/dists/ubuntu **lucid main.**
- Mettez à jour votre cache de dépôts :

sudo apt-get update

Puis exécutez la commande suivante pour installer la dernière version de TinyOs et tous ses outils pris en charge :

sudo apt-get install tinyos

Cela vous donnera probablement un message vous invitant à choisir entre les deux versions disponibles. Un exemple à exécuter ensuite est :

Suivante à votre fichier ~ /. Bashrc ou ~ /.profil dans votre répertoire home afin de mettre en place l'environnement pour le développement TinyOS lors de la connexion.

#Sourcing the tinyos environment variable setup script
source /opt/tinyos-2.1.1/tinyos.sh

Si vous exécutez généralement TinyOS à partir de CVS et nécessitent seulement l'installation de **toolchain**, vous pouvez installer le paquet **tinyos-required** au lieu de**tinyos**.

Pour l'instant il est préférable de supprimer tous les anciens paquets **TinyOs** avant d'installer les nouveaux. Aussi, vous avez utilisé le référentiel TinyOs Debian dans le passé, garder à l'esprit que tous les outils ont été mis à jour pour TinyOs-2.1.1, mais toujours travailler avec toutes les anciennes versions de TinyOs.

Annexe B : Programmation des capteurs

Branchez le capteur Telos B à votre ordinateur avec un câble USB et ouvrez une console sous UBUNTU. Avant de compiler et d'installer le programme, il faut tester si le capteur a été bien détecté et reconnu par le système. Entrez la commande suivante pour lister tous les capteurs branchés :

root@amel :~$ motelist

Si tout se passe bien, le système va vous retourner les informations du capteur branché :

```
Reference   CommPort        Description
----------- --------------- -------------------------
UCC89MXV    /dev/ttyUSB2    Telos (Rev B 2004-09-27)
```

Voilà, il a bien détecté qu'un capteur Telos B est branché sur le port USB numéro 2.

Quand vous testez cette commande sur votre machine, le numéro pourrait être différent (0, 1, 3 etc.). Maintenant on peut compiler et installer le programme à l'aide de Makefile fourni :

root@amel :~$ make telosb install /dev/ttyUSB2

La traduction de cette commande est : Compiler le programme pour la plateforme TelosB, et l'installer vers le port /dev/ttyUSB2.

Maintenant pour lancer l'application on tape la commande suivante:

root@amel :~$ java net.tinyos.tools.MsgReader TemperatureMsg –comm serial@/dev/ttyUSB0:telosb

Bibliographie

Références bibliographiques

[1] AKYILDIZ (I. F.), SU (W.), SANKARASUBRAMANIAM (Y.) et CAYIRCI. (E.), "Wireless sensor networks: a survey", *IEEE Communications Magazine*, vol. 40, no 8, p. 102–114, August 2002.

[2] Julien BEAUDAUX, « Partitionnement logique dans les réseaux de capteurs sans fil», Mémoire de Master, Université de Strasbourg, Laboratoire des Sciences de l'Image de l'Informatique et de la Télédétection, 2010.

[3] LEHSAINI Mohamed, «Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique », Thèse de Doctorat, Université A.B Tlemcen et Université de Franche-Comté, Tlemcen, 2009.

[4] DHIBEYA, «Routage avec QoS temps réel dans les réseaux de capteurs », Rapport de projet de fin d'études, École supérieure des communications, Tunis, 2007.

[5] Yacine CHALLAL, Hatem BETTAHAR, Abdelmadjid BOUABDALLAH, «Les Réseaux de capteurs (WSN: Wireless Sensor Networks) », Rapport interne, Université de Technologie de Compiègne, France, 2008.

[6] MAKHOUL Abdallah, « Réseaux de capteurs : localisation, couverture et fusion de données ».Thèse de Doctorat, Université de Franche-Comté, 2008.

[7] Claude CASTELLUCCIA, « La Sécurité des Capteurs et Réseaux de Capteurs », Projet Planete, INRIA, Juin 2008.

[8] Mathieu BADNET et Nicolas BELLOIR «Réseaux de capteurs : Mise en place d'une plate forme de test et d'expérimentation», Mémoire de Master Technologie de l'Internet, France, 2005/2006.

[9] Ali MAHMOUD, Annette BOHM and Magnus JONSON. "Wireless sensor networks for surveillance applications - A comparative survey of MAC protocols". In Proceedings of the 4th International Conference on Wireless and Mobile Communications (ICWMC'08), pp399–403, Washington, DC, USA, 2008.

[10] John Paul WALTERS et al.,"Wireless Sensor Network Security: A Survey", Department of Computer Science Wayne State University, 2006.

[11] Boucif Amar BENSABAR,« Introduction à la sécurité des réseaux de capteurs sans fil » ;Cours,laboratoire de Mathématiques et Informatique Appliquées LAMIA,Uni versité du Québec à Trois-Rivières.

[12] Du W., Deng J., Han, Y. S. and P. Varshney , «A witness-based approach for data fusion assurance in wireless sensor networks», In IEEE Global Communications Conference (GLOBECOM), Vol. 3, pp. 1435–1439, 2003.

[13] HOBLOS G., STAROSWIECKI M., AITOUCHA A. « Sur la Tolérance aux Fautes de Capteurs et d'Actionneurs », Journal Européen des Systèmes Automatisés, vol. 35, n°3, 331-352, 2001.

[14] KACIMI Rahim, « Techniques de conservation d'énergie dans les réseaux de capteurs sans fil », Thèse de Doctorat, Université de Toulouse, 2009.

[15] Ioannis Parissis LCIS & Claudia Roncancio LIG « ARTEcoApplications à base de Réseaux de capteurs Tolérantes aux Fautes &Energétiquement économiques », Mémoire, Grenoble Institue of Technology, France, 2011.

[16] Thibault BERNARD, « Marches aléatoires et mot circulant adaptabilités et tolérance aux pannes dans les environnements distribués », Thèse, Université de Reims-Champagne-Ardenne, 2011.

[17] Azzedine Boukerche, Richard Werner Nelem Pazzi, Regina Borges Araujo "A Fast and Reliable Protocol for Wireless Sensor Networks in Critical Conditions Monitoring Applications*", paper : MSWiM'04 Copyright 2004 ACM 1-58113-953-5/04/0010,Canada, 2004.

[18] Habib M. Ammari, « Challenges and Opportunities of Connected K-Covered Wireless Sensor Networks », Livre, 2009.

[19] François Taïani, Marc-OlivierKillijian, Jean-Charles Fabre, « Intergiciels pour la tolérance aux fautes Etat de l'art et défis », Revue des sciences et technologies de l'information, série TSI, Éditions Hermès Lavoisier, Vol. 25, n. 5, juin-juillet 2006, pp. 599-630 (32p.), doi: http://dx.doi.org/10.3166/tsi.25.599-630.

[20] Crossbow. MICA2 Data sheet. [Online] 2009.