

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaïa

Faculté des Sciences Exactes

Département d'Informatique



جامعة بجاية
Tasdawit n' Bgayet
Université de Béjaïa

Mémoire de fin de cycle
en vue d'obtention du diplôme de master recherche en informatique
spécialité : Réseaux et systèmes distribués

Thème

La Gestion des Données dans les Réseaux de Capteurs sans Fil.

Présenté par :

M^{elle} HAMOU ALDJA Kahina

M^{elle} HARKATI Samia

Devant le jury composé de :

Président	M ^r SALHI Nadir	MAA Université de Béjaïa
Examineur	M ^{me} BOUTRID Samia	MAB Université de Béjaïa
Examineur	M ^{me} HOUARI Rima	MAB Université de Béjaïa
Encadreur	M ^r SEBAA Abderrazak	MAA Université de Béjaïa
Co-Encadreur	M ^{elle} KHOULALENE Nadjjet	MAA Université de Béjaïa

Promotion 2012/2013

Remerciements

En premier lieu nous remercions Dieu le tout puissant pour toute la volonté et le courage qu'il nous a donné pour l'achèvement de ce travail.

Nous ne saurons trouver les termes qu'il faut pour exprimer notre profonde gratitude et la reconnaissance que nous devons à notre promoteur M^r SEBAA pour la qualité de l'encadrement, dont il nous a fait bénéficier, pour avoir mieux guidé et structurer ce travail en conjuguant habilement, disponibilité, conseils et critiques constructives. Ainsi que de nous avoir fait profiter de son expérience.

Nous tenons à remercier très chaleureusement notre co-promotrice M^{elle} KHOULALENE pour son aide judicieuse, sa disponibilité, ses orientations et ses précieux conseils. L'aboutissement de ce travail doit beaucoup à sa contribution.

Toutes nos sincères gratitudee et notre profond respect à M^r SALHI qui nous a honorés en acceptant de présider notre soutenance.

Nous remercions vivement M^{me} BOUTRID et M^{me} HOUARI qui ont accepté d'examiner et de valoriser notre travail.

On remercie toutes les personnes qui nous ont aidé durant la préparation de ce mémoire de près ou de loin surtout HAROUN Lamine.

Un grand merci à tous nos collègues en Master 2 et tous les enseignants du département Informatique, qui nous ont offert un environnement de travail extrêmement agréable.

Un grand merci à notre famille, pour son soutien qui nous a poussé à chercher au fond de nous la volonté de faire toujours beaucoup plus, à nos amis et tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

Dédicaces

Je dédie ce modeste travail à mes chers parents

aucune dédicace ne saurait être assez éloquente pour vous exprimer ce que vous méritez pour tous les efforts et les sacrifices que vous n'avez jamais cessé de consentir pour mon instruction et mon bien-être.

À mes frères : Karim, Farid, Hakim et Nacer

mes sœurs :Kahina, Lamia et Dyhia, ma nièce Iliana

Les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je vous porte.

À toute ma famille.

À mon binôme Kahina.

À tous mes ami(e)s :Nawel, Nesma, Thili, Hayet, Taous, Dalal, Fawzi, Cherif, yougou

et ceux qui me sont chers, la liste est bien longue.

HARKATI Samia

Dédicaces

Je dédie ce travail à mes chers parents

Hamid et Farida

*qui m'ont donné un magnifique model de labeur et de persévérance tout au long
de mon cursus et pour m'avoir apporté un grand support moral lors de la
rédaction de ce mémoire.*

*À mes frères :Juba, Kossaila et Syphax, pour leur aide morale et leur simple
présence à mes côtés.*

À toute ma famille.

À mon binôme Samia.

À tous mes ami(e)s.

HAMOU ALDJA Kahina

TABLE DES MATIÈRES

Table des Matières	i
Liste des tableaux	iii
Table des figures	iii
Liste des abréviations	iv
1 Généralités sur les réseaux de capteurs	1
1.1 Introduction	1
1.2 Définition et présentation des réseaux de capteurs	1
1.2.1 Les capteurs sans fil	1
1.2.2 Les réseaux de capteurs sans fil	2
1.3 Architecture d'un nœud capteur	2
1.3.1 Architecture matérielle	2
1.3.2 Architecture logicielle	4
1.4 Architecture d'un réseau de capteur	5
1.4.1 Architecture matérielle	5
1.4.2 Architecture protocolaire	6
1.5 Caractéristiques des réseaux de capteurs	7
1.5.1 Déploiement	8
1.5.2 Durée de vie	9
1.5.3 Accès sans fil (L'interférence)	9
1.5.4 Topologie du réseau	9
1.5.5 Ressources limitées	10
1.5.6 Bande passante limitée	10

1.6	Le routage dans les réseaux de capteurs	11
1.6.1	Le routage axé sur les données (data-centric)	11
1.6.2	Le routage hiérarchique	11
1.6.2.1	Election de cluster-head	12
1.6.2.2	Communication intra-cluster et inter-cluster	12
1.6.2.3	Maintenance des clusters	13
1.7	Domaines d'application des RCSF	13
1.7.1	Applications militaires	14
1.7.2	Applications environnementales	14
1.7.3	Applications médicales	15
1.7.4	Agriculture de précision	16
1.7.5	Applications domestiques	16
1.8	Conclusion	17
2	Les données dans les Réseaux de Capteurs sans Fil	18
2.1	Introduction	18
2.2	Concepts	18
2.2.1	La gestion des données collectées	18
2.2.2	Les requête dans les réseaux de capteurs	21
2.2.2.1	Les types de requêtes possibles	21
2.2.3	Compression de données dans les réseaux de capteurs	21
2.2.4	La diffusion	22
2.3	L'agrégation des données	23
2.3.1	Problématique d'agrégation	23
2.3.2	Définition de l'agrégation de données	23
2.3.3	Les mécanismes d'agrégation	24
2.3.4	Les techniques d'agrégation	25
2.3.5	Les protocoles d'agrégation	27
2.3.5.1	Les protocoles d'agrégation dans les clusters	27
2.3.5.2	Les protocoles d'agrégation dans un arbre	29
2.4	Conclusion	30
3	Approches de gestion de données dans les RCSFs avec les vues matérialisées	31
3.1	Introduction	31
3.2	Définitions et concepts	31
3.2.1	Vues	32
3.2.2	Vues matérialisées	32

3.2.3	Maintenance des vues matérialisées	32
3.2.4	Ensemble de limites k-couvert (V'_i)	33
3.2.5	Requêtes d'agrégation spatiale	33
3.2.6	Fenêtre des requêtes	33
3.2.7	Réseau de capteurs fédéré	33
3.2.8	Nœud consommateur et nœud producteur	34
3.3	Approches de gestion de données dans les RCSFs avec les vues matérialisées . .	34
3.3.1	L'approche de vue matérialisée pour maintenir les opérations d'agrégation à travers de longues périodes dans les réseaux de capteurs	34
3.3.2	Les vues MINT : Les vues matérialisées top-k distribuées	36
3.3.3	Traitement des requêtes d'agrégation multiples dans les réseaux Geo- capteur	38
3.3.3.1	Analyse complète	38
3.3.3.2	Avantages	39
3.3.3.3	Inconvénient	39
3.3.3.4	Réplication du Cluster	39
3.3.3.5	Somme préfixe	39
3.3.4	Approche basée sur les modèles	41
3.4	Analyse et comparaison	44
3.5	Conclusion	46
4	La proposition	47
4.1	Introduction	47
4.2	Hypothèses et exigences	47
4.2.1	Hypothèses de travail	47
4.2.2	Exigences	48
4.3	Présentation du système	49
4.3.1	Les différents types de nœuds du système	49
4.3.1.1	Le capteur (<i>sensor</i>)	49
4.3.1.2	L'agrégateur (<i>aggregator</i>)	49
4.3.1.3	Le puits	49
4.4	Définition des symboles	50
4.5	Présentation générale de l'approche de segmentation de données	50
4.6	Description détaillée de l'approche de segmentation de données	51
4.6.1	La phase de pré-exécution	51
4.6.2	La phase d'exécution	51
4.7	Amélioration de l'approche proposée	53

4.7.1	Le principe de l'amélioration	53
4.8	Déroulement de l'algorithme de l'approche de segmentation de données	54
4.8.1	Présentation du scénario	54
4.8.2	Déroulement de l'exemple sans l'algorithme proposé	55
4.8.3	Déroulement de l'exemple avec l'algorithme de segmentation de données .	56
4.8.4	Déroulement de l'algorithme de l'approche améliorée	59
4.9	Conclusion	61
5	Simulation et Evaluation de performances	62
5.1	Introduction	62
5.2	Métriques de performances	62
5.3	Modèle de simulation	63
5.3.1	Choix du langage de programmation	63
5.3.2	Description du système	63
5.3.3	Modèle d'énergie	64
5.3.4	Etapas de réalisation du simulateur	65
5.4	Evaluation de performances	66
5.4.1	Nombre de messages	66
5.4.2	Energie consommée	67
5.4.3	Durée de vie	68
5.5	Conclusion	69

LISTE DES TABLEAUX

3.1	les formules de la maintenance des vues incrémentales.	36
3.2	Tableau de comparaison entre les quatre approches	46
4.1	Définition des symboles	50
4.2	Les résultats de la collecte sans l'algorithme de segmentation de données	56
4.3	Les classes de données	57
4.4	Les classes de donnée de type de données humidité	60

TABLE DES FIGURES

1.1	Capteur sans fils [3].	2
1.2	Architecture matérielle d'un capteur [2].	4
1.3	réseau de capteurs sans fils [11].	6
1.4	Modèle en couches du réseau de capteurs sans fil [13].	7
1.5	déploiement aléatoire [44].	9
1.6	Topologies des réseaux de capteurs sans fil [17].	10
1.7	Configurations des cluster-heads [16].	12
1.8	Communication intra-cluster et inter-cluster [16].	13
1.9	Application militaire [44].	14
1.10	Application environnementale [44].	15
1.11	Application médical [44].	16
1.12	Agriculture de précision [44].	16
2.1	Cycle d'acquisition des données [24].	19
2.2	Arbre d'agrégation de données [24].	20
2.3	Réalisation de la diffusion : (a) propagation de l'intérêt (b) établissement de gradient, (c) envoi des données [32].	23
2.4	Exemple sans agrégation [23].	24
2.5	Exemple avec agrégation [23].	25
2.6	Les techniques d'agrégation de données dans RCSF [23].	25
2.7	Réseau de capteurs à base de cluster. Les flèches indiquent les liens de communication sans fil [25].	26
2.8	l'agrégation de données dans un arbre [26].	27
2.9	Le protocole TEEN [29].	28
2.10	Plan de requête au leader [23].	29

4.1	Organigramme de la phase de collecte de données	53
4.2	Architecture générale de notre domaine d'application.	55
4.3	Circulation des paquets de données pendant la phase de collecte à l'instant T_0 (avec l'algorithme proposé).	58
4.4	Circulation des paquets de données pendant la phase de collecte à l'instant T_1 (avec l'algorithme proposé)	59
4.5	Circulation des paquets de données pendant la phase de collecte à l'instant T_2 (avec l'approche améliorée)	61
5.1	Modèle d'énergie.	64
5.2	Exemple illustratif du prototype	65
5.3	Variation du nombre de messages circulant dans le réseau en fonction des requêtes.	67
5.4	Variation d'énergie résiduelle en fonction des requêtes.	68
5.5	Distribution de la durée de vie de chaque capteur en fonction des requêtes.	68

LISTE DES ABRÉVIATIONS

APTEEN	A daptive de T hreshold-sensitive E nergy E fficient sensor N etwork protocol
CAN	C onvertisseur A nalogique N umérique
GPS	G lobal P ositioning S ystem
HT	H ard T hreshold
LEACH	L ow E nergy A daptive C lustering textbfHierarchy
MINT	M aterialized I n- N etwork T op-k
MINV	M aterialized I n- N etwork V iew
PEGASIS	P ower E fficient G athering in S ensor I nformation S ystems
RCSF	R éseau de C apteur S ans F il
SOS	S ensor O perating S ystem
SQL	S tructured Q uery L anguage
ST	S oft T hreshold
TDMA	T ime D ivision M ultiple A ccess
TEEN	T hreshold sensitive E nergy E fficient sensor N etwork protocol
TiNA	T emporal coherency-aware i n- N etwork A ggregation
TinyOS	T iny O pen S ystem
WSN	W ireless S ensor N etwork

INTRODUCTION GÉNÉRALE

Le besoin d'observer et éventuellement de contrôler des phénomènes physiques tels que la température, la pression ou encore la luminosité est essentiel pour de nombreuses applications industrielles, scientifiques, et même grand public. Cette tâche est déléguée aux capteurs dont la fonction est l'acquisition de l'information sur les phénomènes observés et, le cas échéant, l'exécution des traitements qui s'y attachent.

De nos jours les technologies sans fils font parties de notre quotidien. Que ce soit des simples télécommandes infrarouge à l'internet sans fil en passant par les périphériques Bluetooth comme les téléphones portables, les souris, etc. aucune maison moderne n'échappe à la règle. Nous constatons que de plus en plus de matériel électronique devient communicant, le plus souvent sans fils, et nous ne voyons pas ce qui pourrait freiner cette évolution. C'est ainsi que de nouvelles voies d'investigation ont été ouvertes avec l'émergence des réseaux de capteurs sans fil (RCSFs). Des réseaux à hautes autonomies et à infrastructures non-prédéfinies, utilisés pour observer et éventuellement de contrôler des phénomènes physiques.

Les réseaux de capteurs sans fil (RCSF) sont une thématique de recherche en pleine expansion. Durant les dix dernières années, de nombreux travaux et études ont été menés pour répondre aux différentes problématiques posées. Tout développement réalisé dans les RCSF est tenu de respecter des contraintes en rapport avec les ressources disponibles et le niveau de miniaturisation souhaitée. Ces contraintes ont eu pour conséquence soit de créer des problématiques nouvelles, soit d'amplifier certaines déjà existantes comme, par exemple, celles relatives à la communication sans fil et la gestion de données. Ainsi, tout module de communication sans fil doit, en plus d'assurer la transmission de données, consommer un minimum d'énergie pour pouvoir envisager son utilisation dans un RCSF. De plus, les ressources d'un capteur sans fil que sont l'énergie, la mémoire et la puissance de calcul sont liées

entre elles. Généralement, la préservation de l'une d'entre elles implique une consommation plus importante de l'une ou des deux autres. Le bon fonctionnement de l'application supportée passe donc par une gestion intelligente des ressources présentes au sein de chaque capteur du réseau.

Un réseau de capteurs est généralement dense et gère des données redondantes. La redondance est due au fait que plusieurs capteurs observent une même portion de la zone de déploiement. De ce fait, lorsqu'un événement se produit sur cette dernière, un nombre élevé de capteurs vont acheminer les mêmes données aux stations de base, engendrant ainsi des traitements intermédiaire inutile des messages, et des collisions. Ce qui entraîne un gaspillage de l'énergie. En revanche, une fois déployés, les capteurs sont inaccessibles. Donc, il est impossible de changer ou de recharger les batteries. Doù la contrainte d'énergie.

Pour répondre à cette problématique, nous proposons une approche utilisant la technique d'agrégation des données. Elle consiste à segmenter les données captées, qui donnera lieu à des classes et qui seront stockées dans un buffer de chaque capteur, notre approche consiste à maintenir des vues stockant les données détectées ou reçues et d'envoyer seulement les nouvelles données qui n'appartiennent pas à la classe de la donnée détectée précédemment, cela permet de réduire le nombre de messages transmis par les capteurs et donc pouvoir consommer de l'énergie. Comme on a proposé une amélioration pour cet approche qui suit le même algorithme à la différence que cet approche est proposée dans le cas des capteurs qui détectent des données hétérogènes et qui consiste à profiter d'un envoi d'une donnée pour améliorer la précision des données de natures différentes.

Objectifs de la recherche

L'objectif principal de ce mémoire est de concevoir une solution permettant de collecter les données issues d'un réseau de capteurs sans fils, tout en respectant l'ensemble des contraintes fixées. De manière plus spécifique ce mémoire vise à :

- Analyser la problématique de l'énergie et la mémoire dans les réseaux de capteurs sans fils, ainsi que d'autres approches déjà proposées dans le domaine, pour en extraire les contraintes auxquelles la nouvelle approche devra se soumettre.
- Proposer une nouvelle approche visant à pallier les inconvénients remarqués, et répondant aux contraintes que nous avons fixées.
- Réaliser une simulation de l'approche afin de valider la faisabilité de l'approche proposée.

Organisation Du mémoire

Pour mener à bien notre travail, nous l'avons organisé en cinq chapitres selon un plan méthodologique suivant :

Le premier chapitre constitue une introduction au domaine très vaste des réseaux de capteurs. Nous présentons les différents concepts liés à la mise en œuvre d'un réseau de capteurs, ses caractéristiques et les différents domaines d'applications.

Le deuxième chapitre décrit, les principales problématiques de recherche dans les réseaux de capteurs telles que la collecte et la gestion de données, l'agrégation des données etc. Ainsi nous présentons les différents protocoles de routage liées à l'agrégation de données.

Dans le troisième chapitre, après les définitions de quelques concepts, nous avons étudié dans l'état de l'art quelques approches de gestion de données dans les réseaux de capteurs basées sur les vues matérialisées.

Le quatrième chapitre présente notre contribution dans la problématique d'optimisation de l'utilisation des ressources des capteurs (énergie, mémoire, processeur) dans les réseaux de capteurs sans fil, en décrivant les détails conceptuels et l'algorithme proposé.

Les résultats de simulation obtenus sont présentés et discutés dans le cinquième chapitre.

Enfin, notre mémoire s'achève par une conclusion générale résumant les grands points qui ont été abordés ainsi que des perspectives que nous souhaitons accomplir prochainement.

CHAPITRE 1

GÉNÉRALITÉS SUR LES RÉSEAUX DE CAPTEURS

1.1 Introduction

Les progrès réalisés ces dernières décennies dans les domaines de la microélectronique, de la micromécanique, et des technologies de communication sans fil, ont permis de produire à un coût raisonnable des composants de quelques millimètres cubes de volume. De ce fait, un nouveau domaine de recherche s'est créé pour offrir des solutions économiquement intéressantes et facilement déployables à la surveillance à distance et au traitement des données dans les environnements complexes et distribués : les réseaux de capteurs sans fil. Les réseaux de capteurs sans fil sont constitués de nœuds déployés en grand nombre en vue de collecter et transmettre des données environnementales vers un ou plusieurs points de collecte, d'une manière autonome.

L'objectif de ce chapitre est de faire une description synthétique des réseaux de capteurs, leurs architectures, leurs caractéristiques et contraintes ainsi que leurs domaines d'applications variés.

1.2 Définition et présentation des réseaux de capteurs

1.2.1 Les capteurs sans fil

Un capteur est un dispositif qui transforme une grandeur observée (température, pression, humidité, etc.) en une grandeur utilisable (intensité électrique, position d'un flotteur etc) [2].

Un nœud capteur sans fil est un composant physique, de petite taille, capable d'accomplir trois tâches complémentaires : le relevé d'une grandeur physique, le traitement de l'information, et la communication avec d'autres nœuds capteurs.

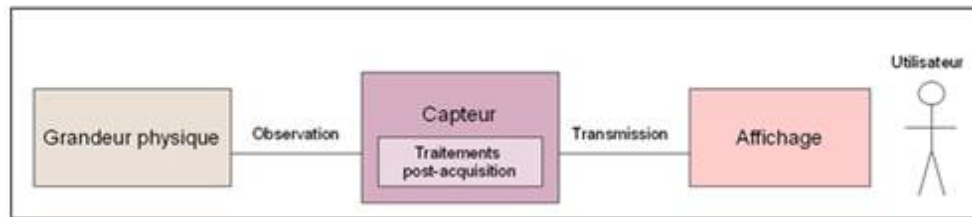


FIGURE 1.1 – Capteur sans fils [3].

Les nœuds capteurs, déployés en grand nombre, sont capables de récolter et de transmettre d'une manière autonome des données de leur environnement immédiat. La localisation géographique de ces nœuds dans le réseau n'est pas obligatoirement prédéterminée. Ils sont déployés dans une zone géographique, appelée champ de capture, définissant le terrain d'intérêt pour le phénomène capturé [3].

1.2.2 Les réseaux de capteurs sans fil

Les réseaux de capteurs sont souvent composés d'un nombre très important de nœuds, appelés capteurs. Déployés de façon à couvrir un territoire donné, les capteurs communiquent par radio afin d'envoyer l'information sur une station collectrice située au cœur ou en bordure du territoire, appelée nœud "Sink". Ces entités doivent être équipées d'une batterie qui est une source d'énergie non durable, ce qui présente la contrainte la plus gênante pour la survie d'un tel réseau. Les réseaux de capteur ne définissent pas une architecture préexistante puisque les capteurs peuvent se trouver fixés à un endroit précis ou mobile, formant ainsi un réseau sans infrastructure prédéfinie. Un tel réseau ne se limite pas à un domaine particulier mais il peut être utilisé dans divers secteurs comme la biologie, la chimie, l'environnement, ainsi que la surveillance sismique et même la télésurveillance personnelle [4].

1.3 Architecture d'un nœud capteur

1.3.1 Architecture matérielle

Suivant le type d'application, il existe une multitude de capteurs sur le marché : les capteurs de température, d'humidité, de pression, etc. Cependant malgré cette diversité apparente, il reste doté d'une architecture matérielle similaire. Un capteur est composé principalement

d'une unité de : captage, traitement, communication et d'énergie [2].

- **Unité de capture d'informations** : elle est composée du capteur proprement dit et du CAN qui transforme les signaux analogiques en signaux numériques. En effet, le capteur observe un phénomène et fournit les signaux analogiques correspondant au phénomène au convertisseur analogique/numérique, le CAN, qui les transforme en signaux numériques compréhensible par l'unité de traitement [5].
- **Unité de traitement d'information** : cette unité constitue l'élément central du capteur. Elle est composée d'un processeur et d'une mémoire intégrant un système d'exploitation spécialement conçu pour les capteurs dont le plus connu est TinyOS. Cette unité possède deux interfaces, une interface pour l'unité d'acquisition et une interface pour l'unité de communication. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à une unité de communication. Cette unité est chargée aussi d'exécuter les protocoles de communication qui permettent de faire collaborer le capteur avec d'autres capteurs. Elle peut aussi analyser les données captées [5], [6].
- **Unité de communication "transceiver unit"** : elle est composée d'un émetteur/récepteur (module radio) permettant la communication entre les différents nœuds du réseau [7].
- **Unité d'énergie** : elle fournit l'énergie nécessaire au fonctionnement du dispositif. Cependant, les dispositifs embarqués étant de petites tailles, les batteries sont aussi petites et donc les ressources énergétiques sont limitées. Cette limitation fait que la gestion énergétique est un point critique pour le capteur et ainsi pour les réseaux de capteurs. Il faut aussi noter que la taille des batteries facilite le transport et fait que ces systèmes sont de plus en plus mobiles [5].

Des composants additionnels peuvent être ajoutés selon le domaine d'application, comme par exemple un système de localisation tel qu'un GPS (Global Positioning System), un générateur d'énergie (exemple : cellules solaires), ou un mobilisateur lui permettant de se déplacer. Ces éléments principaux et optionnels (représentés par des traits discontinus) sont visible sur la figure 1.2 [2].

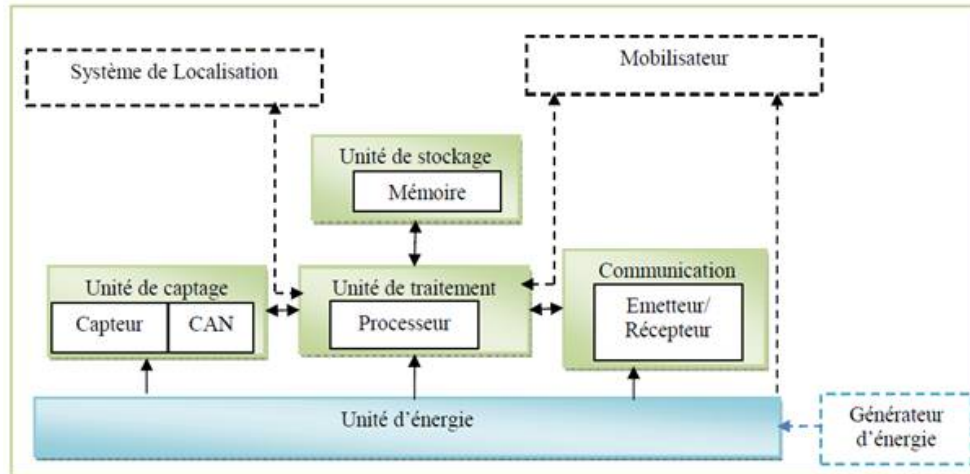


FIGURE 1.2 – Architecture matérielle d'un capteur [2].

1.3.2 Architecture logicielle

Les systèmes embarqués sont des systèmes d'exploitation intégrés, modulaires, destinés aux réseaux de capteurs miniatures. Plusieurs systèmes d'exploitation sont conçus pour les nœuds de RCSF. Parmi ces systèmes, nous citons TinyOS, SOS [7]. En effet, TinyOS est le plus répandue OS pour les réseaux de capteurs sans-fil. Cette plate-forme logicielle ouverte et une série d'outils développés par l'Université de Berkeley est enrichie par une multitude d'utilisateurs. Il est utilisé dans les plus grands projets de recherches sur le sujet (plus de 10.000 téléchargements de la nouvelle version). Le domaine de l'embarqué impose de sévères contraintes notamment en ce qui concerne les problèmes inhérents de l'espace de stockage et l'espace mémoire alloués au système d'exploitation et aux applications tournant dessus. Cet OS répond à ce problème en intégrant très rapidement les innovations en relation avec l'avancement des applications et des réseaux eux même tout en minimisant la taille du code source. Sa conception a été entièrement réalisée en NesC [7], langage orienté composant proche du C, et la bibliothèque de composants de TinyOS est particulièrement complète puisqu'on y retrouve des protocoles réseaux, des pilotes de capteurs et des outils d'acquisition de données. Un programme s'exécutant sur TinyOS est constitué d'une sélection de composants systèmes et de composants développés spécifiquement pour l'application à laquelle il sera destiné (mesure de température, du taux d'humidité...). TinyOS s'appuie sur un fonctionnement événementiel, c'est-à-dire que le capteur ne devient actif qu'à l'apparition de certains événements, par exemple l'arrivée d'un message radio. Le reste du temps, le capteur se trouve en état de veille, garantissant une durée de vie maximale connaissant les faibles ressources énergétiques des capteurs. Ce type de fonctionnement permet une meilleure adaptation à la nature aléatoire de la communication sans fil entre capteurs [8], [9], [10].

1.4 Architecture d'un réseau de capteur

1.4.1 Architecture matérielle

Habituellement les réseaux de capteurs sans fils sont construits autour des quatre principales entités suivantes :

- **Le capteur** : comme le dit bien son nom, il est en charge de mesurer une valeur relative à son environnement (température, pression, luminosité, présence, etc.). On peut parfois rencontrer des capteurs-actionneurs qui non seulement mesureront mais auront aussi pour rôle d'entreprendre une action en fonction de la valeur mesurée. L'intelligence nécessaire à la prise de décision quant à l'action à entreprendre peut alors être déportée sur un autre nœud du réseau.
- **L'agrégateur** : il est en charge d'agréger les messages qu'il reçoit de plusieurs capteurs puis de les envoyer en un seul message au puits (sink). Cette opération a pour principal but de limiter le trafic sur le réseau et donc de prolonger la durée de vie globale du réseau de capteur. Il correspond généralement à la tête d'une grappe (cluster-head). L'utilisation de grappes offre de nombreux intérêts à tous les niveaux, notamment pour le routage.
- **Le puits** : le puits est le nœud final du réseau. C'est à lui qu'est envoyé l'ensemble des valeurs mesurées par le réseau. Il peut arriver qu'il y'ait plusieurs puits sur un même réseau de capteurs.
- **La passerelle** : la passerelle est un dispositif qui a la particularité d'avoir deux interfaces réseau. Il permet de relier le réseau de capteurs sans fils à un réseau plus traditionnel, typiquement l'internet. En effet, habituellement le réseau de capteurs ne sert qu'à faire remonter les mesures, les applications traitant ces informations étant exécutées sur la machine de l'utilisateur final.

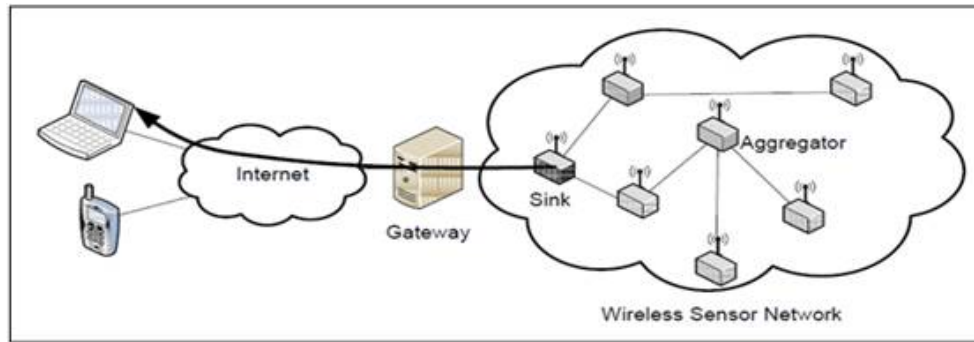


FIGURE 1.3 – réseau de capteurs sans fils [11].

Le fonctionnement global de cette architecture consiste donc à ce que les capteurs fassent des mesures qu'ils font remonter au puits via les agrégateurs. L'application finale tournant sur une machine se situant sur un autre réseau a ainsi accès aux valeurs via une passerelle [11].

1.4.2 Architecture protocolaire

La pile de protocoles utilisée par les nœuds-capteurs est donnée dans la Figure 1.4. Elle combine le routage et la gestion d'énergie et intègre les données avec les protocoles réseau. Elle communique de manière efficace (en termes d'énergie) à travers le support sans fils et favorise les efforts de coopération entre les nœuds-capteurs. La pile de protocoles comprend une couche application, une couche transport, une couche réseau, une couche liaison de données, une couche physique, un plan de gestion d'énergie, un plan de gestion de mobilité et un plan de gestion des tâches. Selon les tâches de détection, différents types de logiciels d'application peuvent être construits et utilisés dans la couche application [12].

1. **Couche physique** : Matériels pour envoyer et recevoir les données,
2. **Couche liaison de données** : Gestion des liaisons entre les nœuds et les stations de base, contrôle d'erreurs ,
3. **Couche réseau** : Routage et transmission des données,
4. **Couche transport** : Transport des données, contrôle de flux,
5. **Couche application** : Interface pour les applications au haut niveau,
En plus des quatre couches, il existe trois plans de gestion à savoir :
6. **Plan de gestion d'énergie** : Contrôle l'utilisation d'énergie,
7. **Plan de gestion de mobilité** : gère les mouvements des nœuds,

8. **Plan de gestion de tâche** : Balance les tâches entre les nœuds afin d'économiser de l'énergie[13].

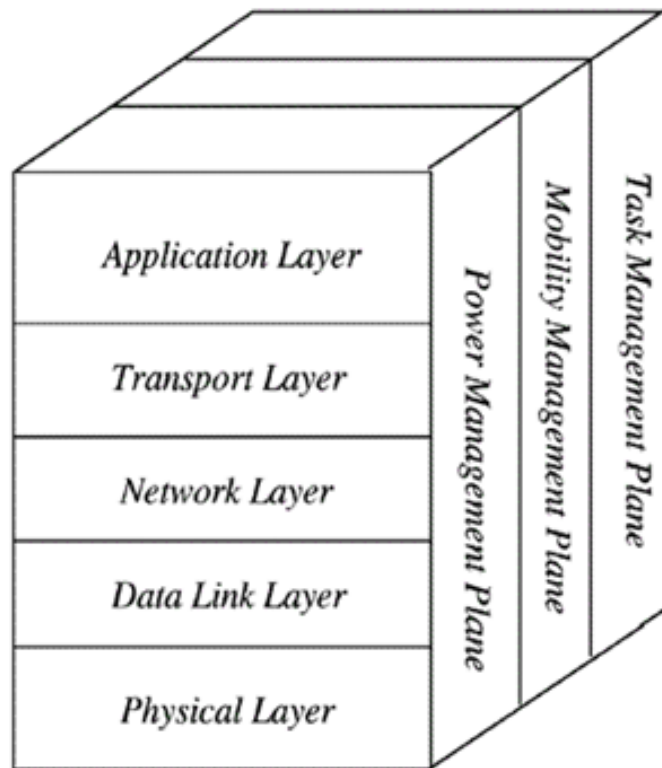


FIGURE 1.4 – Modèle en couches du réseau de capteurs sans fil [13].

En outre, les plans de gestion d'énergie, de mobilité et des tâches surveillent et gèrent la consommation d'énergie, les mouvements, et la répartition des tâches entre les nœuds capteurs. Ces plans aident les nœuds-capteurs à coordonner les tâches de détection et à réduire l'ensemble de la consommation d'énergie. [12].

1.5 Caractéristiques des réseaux de capteurs

L'intégration des réseaux de capteurs avec le monde physique a rendu leur mode de fonctionnement différent de celui des réseaux informatiques traditionnels. Ils possèdent des caractéristiques particulières qui rendent le développement d'applications non-trivial.

Les réseaux de capteurs sans fil sont apparentés aux réseaux ad-hoc[29]. En effet, ces deux types de réseaux ont de nombreux points communs :

- Réseaux sans infrastructure,
- Architecture décentralisée,
- Autonomie,
- Utilisation d'ondes radio pour communiquer.

Bien que de nombreux protocoles et algorithmes ont été proposés pour les réseaux ad-hoc traditionnels, ils ne sont pas bien adaptés aux caractéristiques et exigences des applications.

Les points de différence entre les deux types de réseaux sont :

- La densité des nœuds déployés est beaucoup plus importante dans les réseaux de capteurs,
- Les nœuds capteurs ont des capacités limitées en énergie et mémoire,
- La topologie dans les réseaux de capteurs est souvent dynamique,
- Dans un réseau de capteurs, la communication entre les nœuds se fait par diffusion et non pas point à point,
- Les capteurs peuvent ne pas avoir un identifiant global à cause du grand nombre de nœuds [14].

Dans cette section, nous allons montrer les différentes caractéristiques liées aux réseaux de capteurs telles que le déploiement, la bande passante, la topologie et la connectivité, l'énergie, l'interférence, etc.....

1.5.1 Déploiement

Le déploiement des capteurs est la première opération (phase) dans le cycle de vie d'un réseau de capteurs. On peut envisager plusieurs formes de déploiements selon les besoins des applications.

- **Dispersés aléatoirement**

Les nœuds peuvent être déployés aléatoirement d'un avion ou d'une roquette par exemple, le déploiement aléatoire est adopté dans la majorité des scénarios à cause de raisons pratiques tels que le coût et le temps. Cependant, le déploiement aléatoire ne peut pas fournir une distribution uniforme sur la région d'intérêt, ce qui déclenche de nouveaux problèmes dans les réseaux de capteurs. Les principaux problèmes engendrés sont : la localisation, la couverture de la zone, la connexité et la sécurité [14].



FIGURE 1.5 – déploiement aléatoire [44].

- **Posés à un endroit précis(déploiement déterministe)**

Les nœuds peuvent être placés un par un d'une manière déterministe par un humain ou un robot. Le déploiement peut être fait d'un seul coup ou bien peut être un processus continu en redéployant d'autres capteurs dans une même zone. Dans un grand nombre d'applications, le déploiement manuel est impossible [14].

1.5.2 Durée de vie

L'économie d'énergie est une des problématiques majeures dans les réseaux de capteurs. En effet, la recharge ou le remplacement de batteries est quasi impossible vu les zones d'intérêt qui sont généralement inaccessibles. Il faut donc que les capteurs économisent au maximum leur énergie afin de pouvoir fonctionner longtemps [12].

1.5.3 Accès sans fil (L'interférence)

Les liens radio ne sont pas isolés, deux transmissions simultanées sur une même fréquence, ou utilisant des fréquences proches, peuvent interférer [16].

1.5.4 Topologie du réseau

C'est-à-dire la façon dont le réseau est organisée. Il peut être organisé en simple boucle, en étoile, en graphe (Figure 1.6). Le protocole de communication entre les nœuds et la station de base est alors différent. Dans le cas de la topologie étoile, les nœuds vont uniquement communiquer avec la station de base, alors que dans le cas de la topologie graphe, certains

nœuds vont relayer l'information. Enfin, pour la topologie boucle simple, tous les nœuds peuvent relayer l'information. Le type de topologie sera choisi notamment suivant la taille du réseau, sa densité et l'application visée. [17].

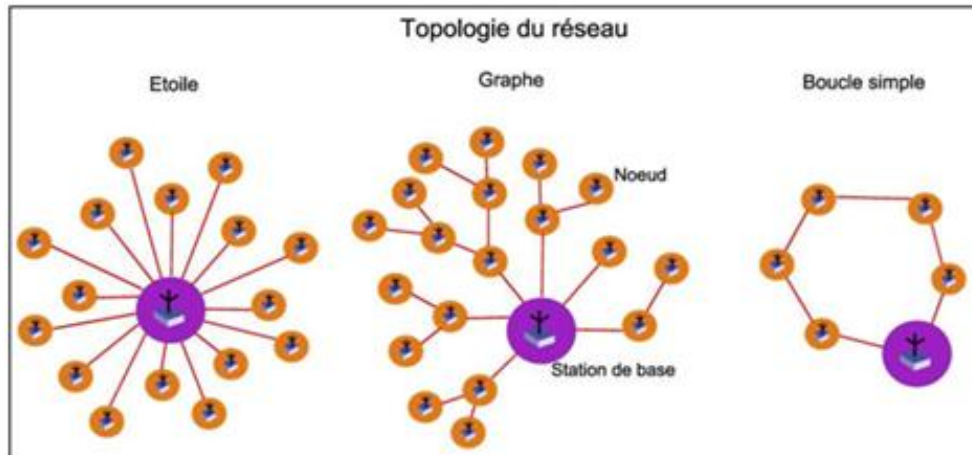


FIGURE 1.6 – Topologies des réseaux de capteurs sans fil [17].

La dynamique du réseau découle des défaillances des nœuds ou des cassures des liens entre ceux-ci. Le réseau de capteurs a une forte dynamique de topologie surtout lorsqu'il s'agit d'un réseau mobile. La moindre défaillance énergétique d'un capteur peut donc changer significativement la topologie du réseau et imposer une réorganisation coûteuse de ce dernier. La perturbation des communications (comme les obstacles, l'interférence, etc.) peut induire des cassures des liens entre les nœuds voisins [14].

1.5.5 Ressources limitées

En plus de l'énergie, les nœuds capteurs ont aussi une capacité de traitement et de mémoire limitée. En effet, les industriels veulent mettre en œuvre des capteurs simples, petits et peu coûteux qui peuvent être achetées en masse [6].

1.5.6 Bande passante limitée

Une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un nœud est limitée [16].

1.6 Le routage dans les réseaux de capteurs

Le routage est un service très important dans les réseaux de capteurs. Il doit permettre l'arrivée des données à la station de base avec le minimum de pertes et de dissipation d'énergie. Les protocoles de routage proposés pour les réseaux de capteurs peuvent être classés en deux grandes familles : le routage plat et le routage hiérarchique [43].

1.6.1 Le routage axé sur les données (data-centric)

Le but principal dans ce routage est de minimiser les redondances de données pouvant devenir très pénalisantes surtout en terme de consommation d'énergie et surcharge du réseau. Dans un tel routage, la station de base diffuse ces requêtes vers des régions particulières ou tout le réseau. Elle attend ensuite, les données correspondant à ses requêtes depuis les noeuds situés dans la région cible ou d'intérêt. Parmi ces protocoles, nous trouvons Directed Diffusion, Minimum Cost Forwarding, Energy Aware Routing [43].

1.6.2 Le routage hiérarchique

La seconde famille regroupe les protocoles qui utilisent différentes techniques d'agglomération (ou clustering) et d'agrégation de données pour minimiser le nombre de messages transmis à la station de base. La formation des grappes et le choix de leurs chefs sont basés sur la réserve d'énergie des capteurs et sur la proximité de ces derniers du chef à élire. Parmi ces protocoles, il y a LEACH, HEED,.....etc [43].

L'approche de clustering consiste à partitionner le réseau en un certain nombre de clusters, plus homogènes selon une métrique spécifique ou une combinaison de métriques, et former une topologie virtuelle. Les clusters sont généralement identifiés par un noeud particulier appelé cluster-head. Ce dernier permet de coordonner entre les membres de son cluster, d'agréger leurs données collectées et de les transmettre à la station de base. Il existe deux configurations possibles pour les clusters : dans la première, les membres du cluster ne communiquent qu'avec le cluster-head, par contre dans la deuxième, ils peuvent utiliser d'autres noeuds comme passerelles vers le cluster-head, comme le montre la figure suivante : [16], [19]

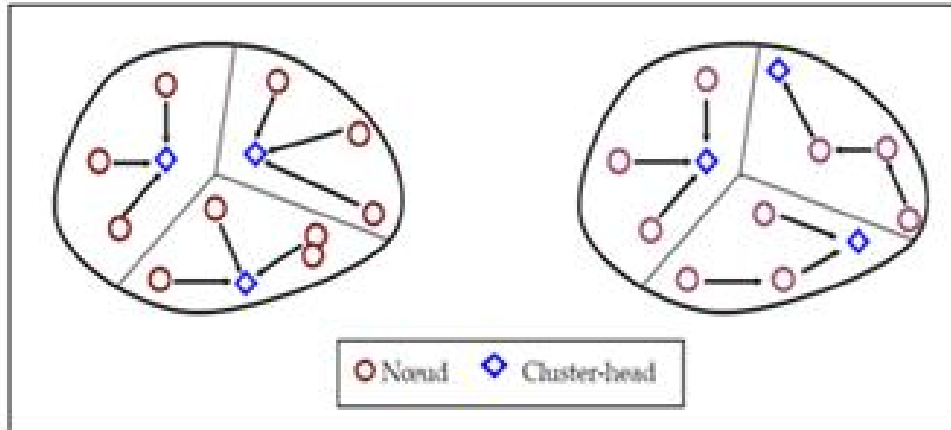


FIGURE 1.7 – Configurations des cluster-heads [16].

1.6.2.1 Election de cluster-head

Il existe plusieurs méthodes de formation de clusters. La plus répandue s'exécute comme suit :

1. Chaque nœud devra connaître son voisinage par le biais des messages Hello,
2. La phase d'élection de cluster-heads appelée aussi la phase Setup utilise une métrique spécifique ou une combinaison de métriques pour chaque nœud telle que le plus grand/petit ID dans son voisinage, le degré de connectivité, la puissance de transmission, l'énergie restante ou la mobilité,.... ou bien un poids qui représente une combinaison de quelques métriques [16],
3. Le nœud choisi comme cluster-head diffuse son statut dans son voisinage et invite ses voisins qui ne sont pas encore affiliés à d'autres clusters de le rejoindre [16].

1.6.2.2 Communication intra-cluster et inter-cluster

Chaque cluster-head se charge des communications à l'intérieur de son cluster et maintient les informations de routage lui permettant de joindre les autres cluster-heads. De plus, comme les cluster-heads ne sont pas directement reliés, des nœuds passerelles peuvent aussi être élus et utilisés pour les communications entre cluster-heads [16].

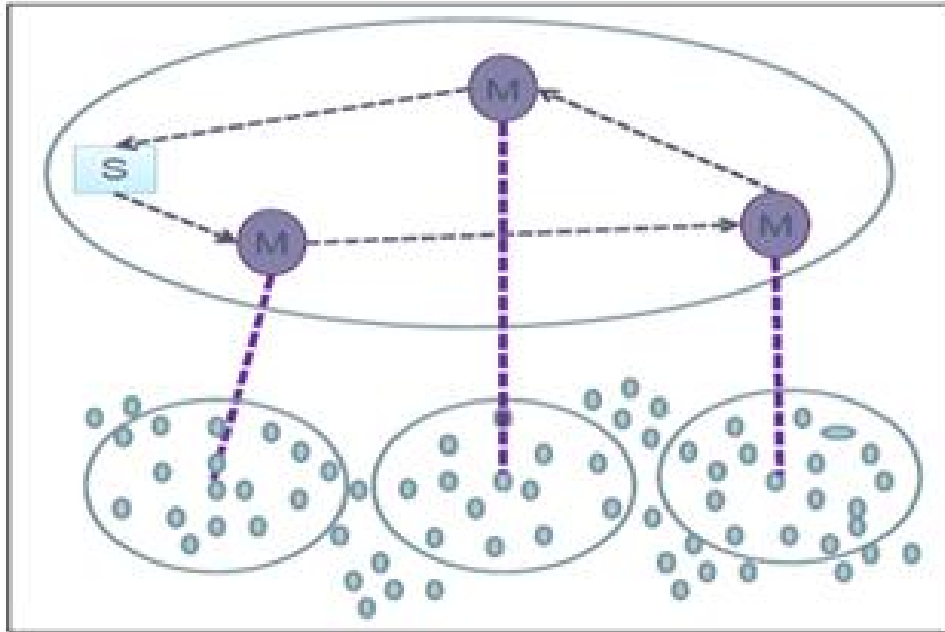


FIGURE 1.8 – Communication intra-cluster et inter-cluster [16].

1.6.2.3 Maintenance des clusters

Dans le but de s'adapter aux changements de la topologie du réseau, une mise à jour des clusters est dynamiquement réalisée dans le cas où un cluster-head ou un membre migre d'un cluster C_i à un autre C_j . D'autre part, si le cluster-head garde son statut le plus longtemps possible, même s'il ne possède pas le poids maximum dans son propre cluster, il perdra son rôle une fois qu'il s'éteindra i.e. sa batterie sera épuisée [16].

1.7 Domaines d'application des RCSF

L'évolution des supports de communication sans fil, la miniaturisation, le faible coût des micro-capteurs et l'élargissement de la gamme des sondes de captage (thermique, humidité, optique, vibrations, etc...) ont élargi le champ d'application des réseaux de capteurs.

Ils permettent de collecter et de traiter des informations complexes issues de l'environnement (météorologie, étude des courants, de l'acidification des océans, de la dispersion de polluants, etc...). Ils s'insèrent également dans d'autres systèmes industriels tels que le contrôle et l'automatisation des chaînes de montage.

Les réseaux de capteurs pourraient aussi révolutionner la manière même de comprendre et de construire des systèmes physiques complexes, notamment dans le domaine militaire, environnemental, domestique, sanitaire, de la sécurité, etc.

1.7.1 Applications militaires

Le domaine militaire a été, comme dans le cas de plusieurs technologies, un précurseur pour le développement d'applications de réseaux de capteurs. Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui rendent ce type de réseaux un outil appréciable dans un tel domaine.

Un réseau de capteurs déployé sur un secteur stratégique ou difficile d'accès, permet par exemple d'y surveiller tous les mouvements (amis ou ennemis), ou d'analyser le terrain avant d'y envoyer des troupes (détection d'agents chimiques, biologiques ou de radiations)[13].



FIGURE 1.9 – Application militaire [44].

1.7.2 Applications environnementales

La détection de mouvements d'oiseaux, d'animaux de petite taille ou d'insectes est une des applications environnementales possibles des réseaux de capteurs sans fil. Il devient ainsi possible d'observer la biodiversité, sans déranger, des espèces animales vulnérables ou difficiles à étudier dans leur environnement naturel, et de proposer des solutions plus efficaces pour la conservation de la faune et de la flore.

Grâce aux réseaux de capteurs, il est également possible de détecter des problèmes environnementaux comme la pollution, les incendies en milieu forestier, ainsi que la possibilité d'évaluer le taux de pesticides dans l'eau.

Sur les sites industriels, les centrales nucléaires ou dans les pétroliers, des capteurs peuvent être déployés en réseau pour détecter en temps réel des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.) afin d'alerter les utilisateurs et secours plus rapidement, pour permettre une intervention efficace [13].

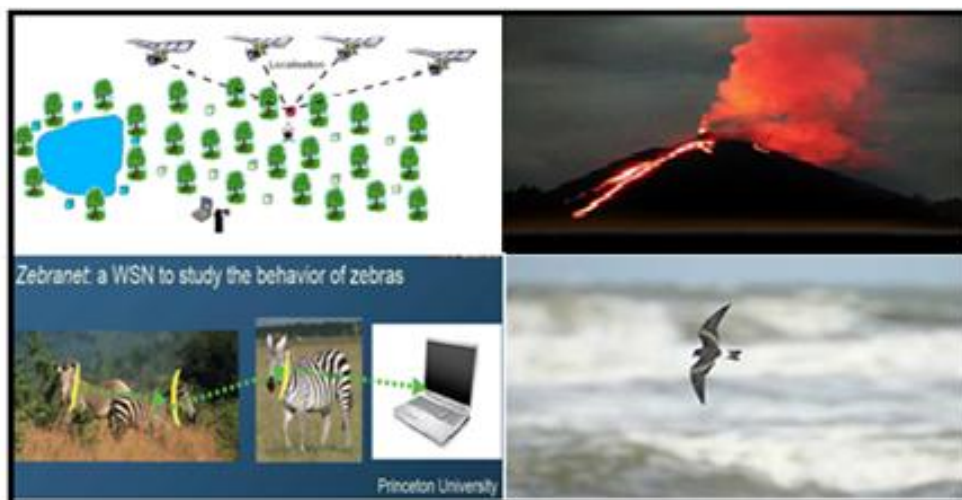


FIGURE 1.10 – Application environnementale [44].

1.7.3 Applications médicales

La surveillance des fonctions vitales d'un organisme vivant peut être facilitée par des micro-capteurs avalés ou implantés sous la peau. Des gélules sous forme de micro-capteurs ou de micro-caméras pouvant être avalées existent déjà et permettent, sans recours à la chirurgie, de transmettre des images depuis l'intérieur d'un corps humain. Une récente étude présente des capteurs pouvant fonctionner dans le corps humain pour le traitement de certaines maladies.

La surveillance de la glycémie, la surveillance des organes vitaux ou la détection précoce de certains cancers sont des applications biomédicales envisagées. Des réseaux de capteurs permettraient également une surveillance temps réel des maternités (détection de vol de bébés) ou des patients [13].

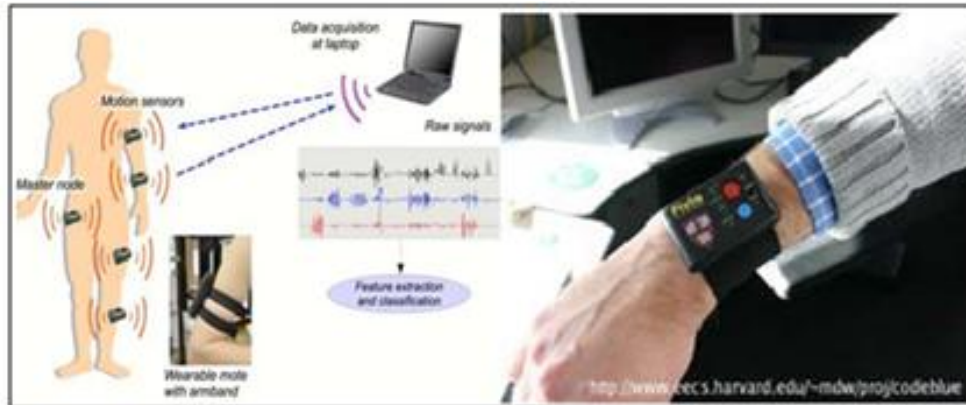


FIGURE 1.11 – Application médical [44].

1.7.4 Agriculture de précision

Les réseaux de capteurs sont capables d'apporter des bénéfices considérables au domaine d'agriculture, grâce à leur habilité de surveiller les taux de pesticides dans l'eau, le degré d'érosion du sol, détection de parasites, et le niveau de pollution de l'air en temps réel [14].



FIGURE 1.12 – Agriculture de précision [44].

1.7.5 Applications domestiques

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, les magnétoscopes, etc. [18]. Ces capteurs embarqués peuvent interagir entre eux et avec un réseau externe via Internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance.

1.8 Conclusion

Dans ce chapitre nous avons présenté les concepts généraux liés aux réseaux de capteurs sans fil. Nous avons décrit le capteur, sa définition et son architecture, ensuite nous avons défini ce qu'un réseau de capteurs, cité ses caractéristiques ainsi que les différents types de communication. Nous avons vu à travers quelques exemples que les réseaux peuvent se révéler très utiles pour de nombreuses applications lorsqu'il s'agit de collecter et traiter des informations provenant de l'environnement.

Dans le chapitre suivant, nous allons poursuivre notre étude sur les réseaux de capteurs en s'intéressant plus particulièrement aux données dans ce type de réseaux, plus précisément la gestion et les mécanismes de traitements des données.

CHAPITRE 2

LES DONNÉES DANS LES RÉSEAUX DE CAPTEURS SANS FIL

2.1 Introduction

Les nœuds capteurs composant le réseau possèdent généralement de faibles capacités de calcul, de mémoire et d'énergie, l'accès au médium radio étant l'élément le plus coûteux. Ainsi agréger les données ou diminuer le nombre de bits transmis afin d'augmenter la durée de vie du réseau est un défi permanent pour ce type de réseau [20], des algorithmes d'agrégation sont utilisés, afin d'obtenir des volumes plus petits à transférer. Nous pouvons rajouter aussi des algorithmes de compression pour assurés une perte tolérable ou minimale d'information [31].

Ce chapitre présente les concepts de base pour le traitement des requêtes dans les réseaux de capteur et la gestion de données collectées, ainsi que et les mécanismes de traitement des données dans les RCSF.

2.2 Concepts

2.2.1 La gestion des données collectées

Les réseaux de capteurs sont destinés à collecter des données sur le monde physique, par conséquent leur utilisation doit être orientée-données. Contrairement aux communications traditionnelles point à point, le routage et la gestion de donnée dans les réseaux de capteurs doivent être traités conjointement afin de pouvoir optimiser la consommation d'énergie. Par conséquent, une composante importante du réseau est de fournir une plate-forme flexible pour

construire des systèmes de gestion de données qui utilisent plusieurs stratégies d'agrégation des données spécifiques pour les différentes applications [14].

La gestion des données collectées rassemble l'ensemble des traitements subis par les données durant leur cycle de vie au sein du RCSF (voir Figure 2.1). L'acquisition à proprement parler est déportée sur le dispositif de mesure de la grandeur physique. Le capteur sans fil est en charge de la bonne récupération des données à la fréquence d'acquisition souhaitée. La réalisation de cette tâche constitue le fonctionnement de base du capteur sans fil auxquelles viennent s'ajouter d'autres, plus ou moins complexes, selon l'application supportée.

La présentation des données se résume à l'envoi des données vers une station centrale de collecte qui affiche les données de manière brute ou sous un format spécifié par l'utilisateur. Dans une version plus élaborée, l'utilisateur peut envoyer des requêtes d'interrogation au RCSF. Il récupère ainsi seulement les données dont il a besoin.

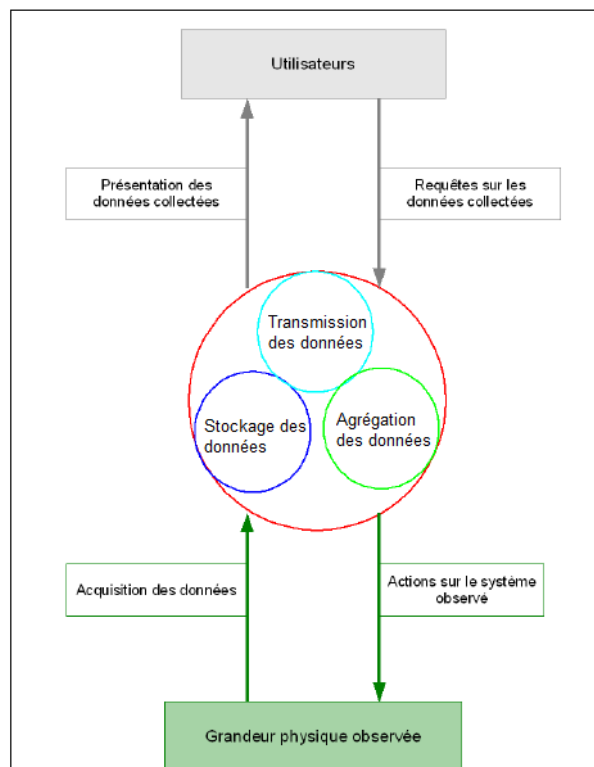


FIGURE 2.1 – Cycle d'acquisition des données [24].

Le stockage de données plutôt que la transmission ne répond pas seulement à une politique d'économie d'énergie. Si les ressources disponibles sont importantes comparativement aux

traitements à effectuer, la solution la plus simple et la plus rapide à mettre en place est une acquisition suivie d'une transmission immédiate des données. Malheureusement, les médias de communication sans fil sont moins fiables que leurs homologues filaires et pâtissent des interférences, des collisions, des multi-trajets causés par le relief montagneux et les bâtiments qui pourraient être situés aux alentours. Ce phénomène est d'autant plus accentué que le déplacement des capteurs sans fil est rapide. En outre, la qualité de service dans les réseaux sans fil est plus complexe à établir et subit d'importantes fluctuations. Par conséquent, durant l'indisponibilité de la communication sans fil, les données devront être stockées. L'emploi d'une mémoire non volatile pourrait être préconisé pour le stockage de données importantes car elles seraient toujours disponibles même en cas de panne d'énergie du capteur.

Une application de détermination de la température moyenne d'une zone donnée illustre parfaitement l'intérêt de l'agrégation de données. Dans un mode de fonctionnement simple, tous les capteurs transmettent à la station centrale de collecte la valeur de température obtenue à l'endroit où ils étaient placés. En intégrant l'agrégation de données, un arbre à différents niveaux est constitué. Les nœuds feuilles de cet arbre sont regroupés en sous-ensemble de capteurs formant des sous-zones ou clusters de premier niveau. Un nœud maître de niveau supérieur est désigné pour chacune de ces sous-zones. Son rôle est de réaliser la première agrégation et de transmettre ensuite ce résultat à son propre nœud maître de niveau supérieur. Ce fonctionnement est répété jusqu'à ce que la station centrale, racine de l'arbre soit atteinte (voir Figure 2.2). Le nombre de messages échangés au sein du réseau s'en trouve ainsi diminué. De plus, les risques de saturation de la station centrale sont limités, le nombre de données à traiter étant très réduit. Le point critique de ce type d'application est l'élection des nœuds maîtres de zone [24].

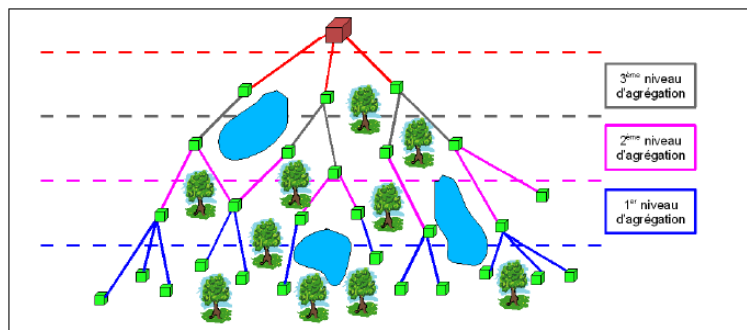


FIGURE 2.2 – Arbre d'agrégation de données [24].

2.2.2 Les requête dans les réseaux de capteurs

Comme dans les systèmes de bases de données traditionnelles, les requêtes décrivent un ensemble logique de données pour lesquelles l'utilisateur est intéressé. Le système peut choisir parmi plusieurs plans pour une requête logique donnée. Par exemple, pour trouver la température moyenne des capteurs du quatrième étage, le système peut collecter des lectures de chaque capteur, puis filtrer la liste des capteurs du quatrième étage et calculer la moyenne. Alternativement, il peut demander qu'aux capteurs du quatrième étage qu'il fournissent leurs températures, et puis faire la moyenne des valeurs collectées. Dans un réseau de capteurs, le dernier plan sera un meilleur choix car il exige seulement des capteurs sur le quatrième étage de collecter et rapporter leurs températures [35].

2.2.2.1 Les types de requêtes possibles

– Les requêtes instantanées

La requête instantanée est évaluée à un temps t d'un ensemble de données disponibles à ce moment. Le résultat est l'ensemble des données satisfaisant la condition de la requête. Deux sous-classes de requêtes instantanées peuvent être distinguées selon les données dont ils ont besoin :

- Requête actuelle : est une requête instantanée qui a évalué sur un ensemble de données décrivant l'état actuel du système surveillé. Le résultat est le jeu de données correspondant à la condition de la requête.
- Requête historique : est une requête instantanée qui a évalué sur un ensemble de données décrivant l'état passé du système surveillé. Le résultat est le jeu de données correspondant à la condition de la requête.

– Les requêtes continues

La requête continue procède sans interruption sur un ensemble de flux de données. Le résultat est un ensemble de flux réponses satisfaisant la condition de la requête [33].

2.2.3 Compression de données dans les réseaux de capteurs

Avec l'évolution des réseaux, la compression de données devient une technique de plus en plus répandue. Les algorithmes de compression se multiplient et deviennent de plus en

plus performants. Le principal objectif d'un algorithme de compression est de réduire la quantité de données pour faciliter son stockage ou son transport. Dans le domaine des réseaux de capteurs, elle constitue une stratégie pour économiser l'énergie en réduisant le nombre de transmissions car c'est l'activité la plus coûteuse pour les capteurs. Aussi, en réduisant la quantité d'information, elle facilite son transport pour ces plateformes qui utilisent des technologies sans fil à faible débit.

Les deux principales opérations dans la compression de données sont : la compression et la décompression. Ces étapes permettent de définir deux catégories de compression : la compression sans pertes et la compression avec pertes. Lorsque l'objet avant compression est identique à celui après décompression, on parle d'un algorithme sans perte sinon lorsqu'il y a une différence entre les deux objets, on parle d'algorithme avec pertes. L'indicateur de performance d'un algorithme de compression s'appelle le taux de compression. La principale raison de l'utilisation d'algorithmes de compression avec pertes est le souci d'amélioration du taux de compression pour des données dont certaines informations peuvent être perdues sans affecter significativement le contenu, par exemple parce que les bits perdus ne représentent que du bruit [5].

2.2.4 La diffusion

L'algorithme, proposé dans [32], est basé sur le paradigme de dissémination de donnée, suivant lequel, le nœud puits diffuse son intérêt, qui présente une description des tâches affectées à tous les nœuds capteurs du réseau, telle que montré dans la figure 2.3(a). Ces descripteurs englobent des paires d'attributs qui décrivent les tâches requises par le nœud puits. Chaque nœud capteur stocke dans son cache une entrée d'intérêts, cette entrée contient un champ d'estampille et d'autres champs de gradients. Comme l'intérêt est propagé à travers le réseau, les gradients à partir du nœud source de donnée vers le nœud puits seront déterminés (figure 2.3(b)). Quand des données seront disponibles au niveau du nœud source, il les envoie au nœud puits suivant le chemin ayant un gradient fort pour l'intérêt diffusé comme illustré dans la figure 2.3(c).

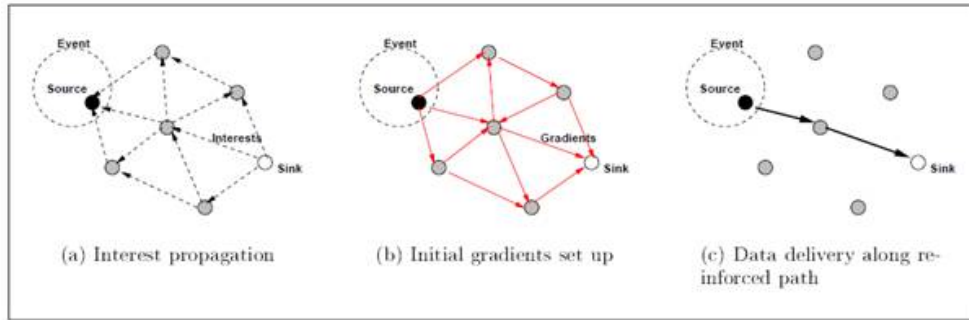


FIGURE 2.3 – Réalisation de la diffusion : (a) propagation de l'intérêt (b) établissement de gradient, (c) envoi des données [32].

2.3 L'agrégation des données

2.3.1 Problématique d'agrégation

Dans un capteur, la problématique principale concerne la consommation d'énergie : Il a été montré dans plusieurs publications scientifiques que la transmission d'un bit est équivalente, en termes d'énergie, à l'exécution d'environ 1000 instructions. Cette valeur augmente avec la portée de la radio. Plus le capteur devra transmettre loin, et par conséquent augmenter sa puissance d'émission, plus il va consommer de l'énergie, et par conséquent réduire sa durée de vie. On estime que la transmission des données d'un capteur représente environ 70% de sa consommation d'énergie. Comme il n'est pas possible de recharger leur énergie ni changer les piles, il est nécessaire d'économiser au maximum l'énergie consommée par ces derniers [1], [23].

De plus dans les RCSF, les données produites par les nœuds capteurs voisins (dense) sont très corrélées spatialement et temporellement. Ceci peut engendrer la réception par la station de base d'informations redondantes. Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau et ainsi d'améliorer sa durée de vie.

2.3.2 Définition de l'agrégation de données

L'agrégation de données est la combinaison des données de différentes sources, et peuvent être mises en application d'un certain nombre de manières, et une des techniques permettant de réduire la quantité de données transférées et, qui permet de réduire la charge de trafic acheminé dans les réseaux. Agrégation des données en général comprend la fusion des données de multiples capteurs à des nœuds intermédiaires et de transmettre des données agrégées à la

station de base [21], [22], [25].

2.3.3 Les mécanismes d'agrégation

L'agrégation de données dans les réseaux de capteurs consiste à remplacer les lectures individuelles de chaque capteur par une vue globale, collaborative sur une zone donnée (clustering).

On peut utiliser par exemple de simples fonctions d'agrégat telles que MIN, MAX ou AVG, qui permettent à partir d'une série de n messages reçus par un " chef de zone " (capteur chef d'une zone) de ne renvoyer vers le puits qu'un seul message résumant l'information contenue dans ces n messages [23].

Par exemple, dans ce réseau de capteurs 18 messages sont envoyés.

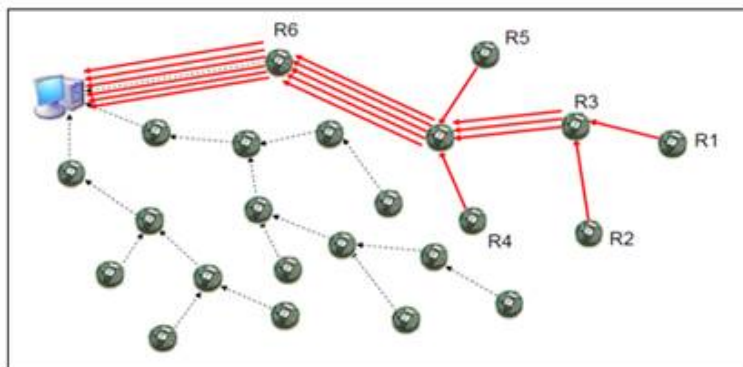


FIGURE 2.4 – Exemple sans agrégation [23].

En utilisant le mécanisme d'agrégation de données, on obtient un total de 7 messages envoyés sur le réseau :

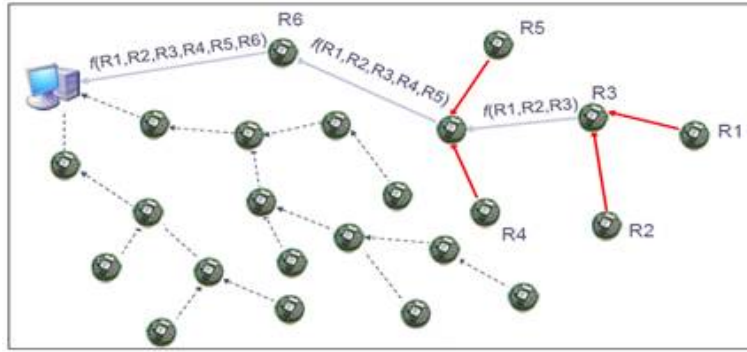


FIGURE 2.5 – Exemple avec agrégation [23].

Moins de messages émis plus économie d'énergie, résultat : maximiser le temps de vie du réseau.

2.3.4 Les techniques d'agrégation

Les techniques d'agrégation de données peuvent être découpées en deux :

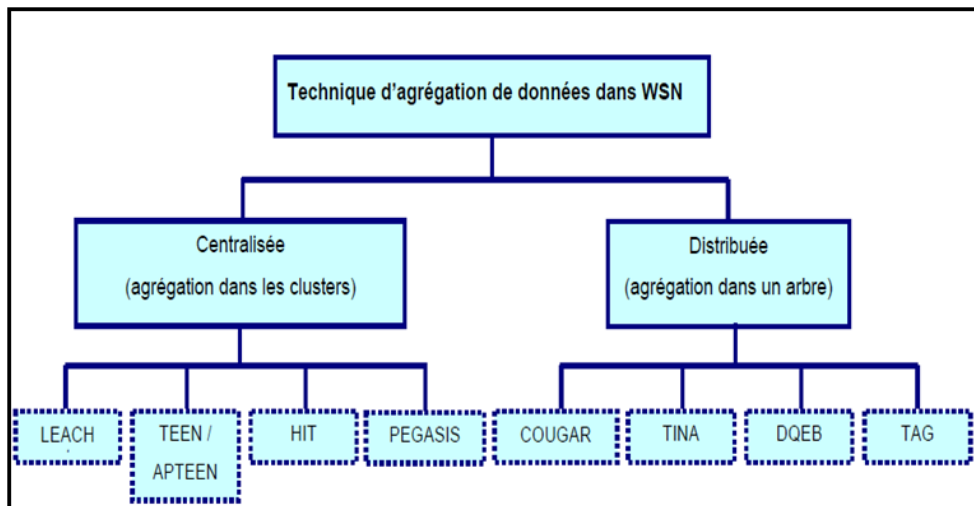


FIGURE 2.6 – Les techniques d'agrégation de données dans RCSF [23].

► Centralisée (agrégation dans les clusters)

Dans les réseaux de capteurs d'énergie contraints de grande taille, il est inefficace pour les capteurs de transmettre les données directement à la station de base. Dans tels scénarios, les nœuds capteurs peuvent transmettre des données à un capteur local "Cluster-head" qui agrège les données de tous les capteurs de son groupe et transmet à la station de

base. La figure 2.7 montre un réseau de capteurs à base de cluster. Le chef de groupe peut communiquer directement avec le sink via des transmissions à longue portée ou plusieurs sauts à travers des cluster-head.

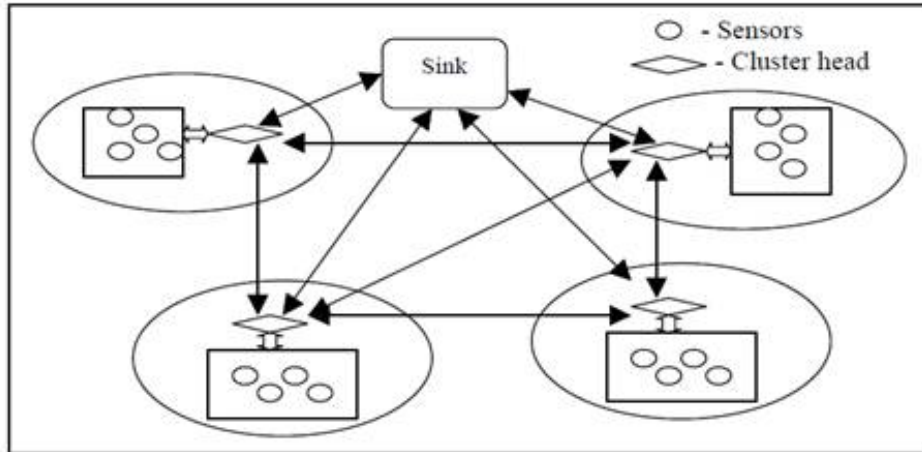


FIGURE 2.7 – Réseau de capteurs à base de cluster. Les flèches indiquent les liens de communication sans fil [25].

► **Distribuée (agrégation dans les arbres)**

Agrégation dans un arbre c'est-à-dire que le réseau est vu de façon globale. En effet, cette technique, utilisée pour résoudre le problème d'implosion de données (Le même message est dupliqué plusieurs fois, chaque nœud reçoit autant de fois la même donnée que le nombre de ses voisins) qui caractérise les réseaux de capteurs, considère ce réseau comme un arbre multicast inversé (figure 2.8) où le nœud puits demande aux nœuds capteurs de reporter les conditions ambiantes du phénomène capté. Les données provenant des différents capteurs sont agrégées si elles décrivent le même attribut du phénomène surveillé, et ceci quand ces données atteignent le même nœud durant leur chemin vers le nœud puits [23], [26].

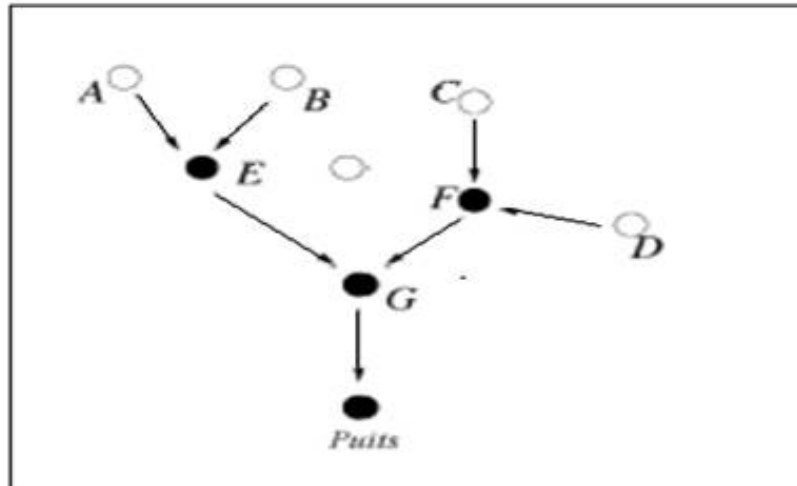


FIGURE 2.8 – l'agrégation de données dans un arbre [26].

2.3.5 Les protocoles d'agrégation

2.3.5.1 Les protocoles d'agrégation dans les clusters

► Low Energy Adaptive Clustering Hierarchy (LEACH)

LEACH est un protocole de routage hiérarchique qui utilise la classification (clustering) afin de diviser le réseau en deux parties : les cluster-heads (chefs de zone) et les nœuds membres. Ce protocole se décompose en deux phases qui se succèdent plusieurs fois : la construction et la communication.

► TEEN/APTEEN

TEEN (Threshold sensitive Energy Efficient sensor Network protocol) est un protocole orienté événements et basé sur le concept de clustering. Chaque nœud capteur possède deux valeurs seuil : HT (Hard Threshold) et ST (Soft Threshold) : la transmission débute lorsque la valeur mesurée excède HT, ou lorsqu'un changement de la valeur mesurée est supérieure ou égale à ST. TEEN suppose que tous les nœuds du réseau ont initialement le même niveau d'énergie et que les cluster-heads ont une source de puissance constante pendant qu'elles doivent attendre avec la survenue d'un événement sur les nœuds capteurs [28].

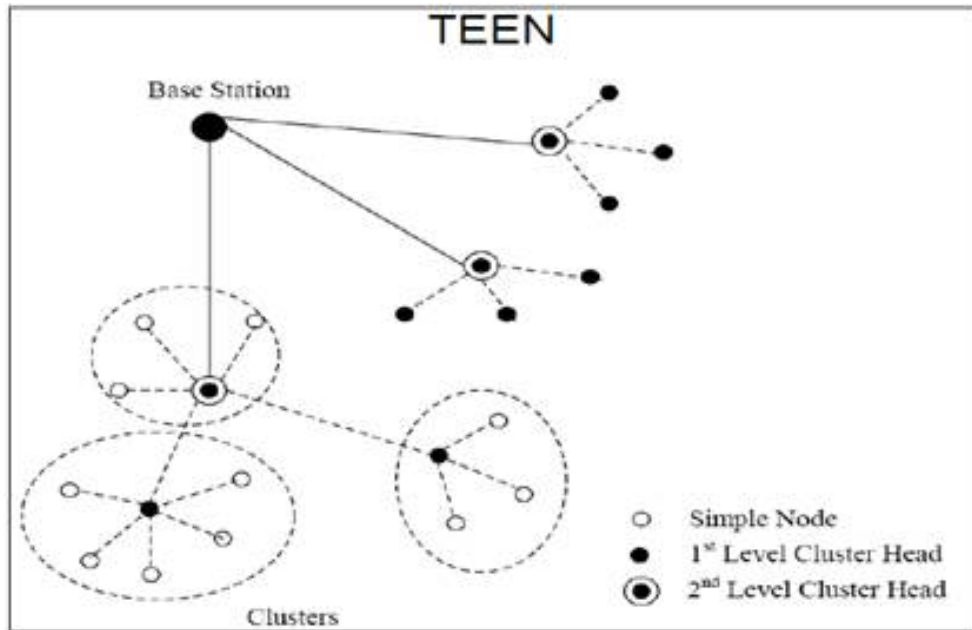


FIGURE 2.9 – Le protocole TEEN [29].

APTEEN (Adaptive Threshold-sensitive Energy Efficient sensor Network protocol) est une extension de TEEN qui vise à l'adapter pour la surveillance périodique, en plus du mode de captage réactif. Le modèle d'architecture adopté par APTEEN est similaire à TEEN, et les seuils HT et ST sont utilisés. Cependant, APTEEN oblige le nœud à capter et à transmettre les données au cluster-head après un délai appelé 'count time'. Pour cela, un slot de temps est associé à chaque membre du groupe suivant un ordonnancement TDMA pour lui permettre de transmettre [29], [30].

► Power Efficient Gathering in Sensor Information Systems (PEGAGIS)

PEGAGIS est un autre protocole de routage conçu pour les réseaux de capteurs, il a été proposé comme amélioration au protocole LEACH. Dans ce protocole, un nœud peut communiquer seulement avec son voisin le plus proche, et doit attendre son tour pour pouvoir transmettre à la station de base. Ceci réduit la quantité d'énergie requise par round pour la transmission des données captés, et par conséquent prolonge la durée de vie du réseau suivant les auteurs de deux comparé à LEACH.

La motivation principale pour ce protocole, consiste à prolonger la durée de vie des nœuds capteurs et réduire la bande passante consommée en utilisant la collaboration locale entre les nœuds et en tolérant la défaillance des nœuds capteurs [26].

2.3.5.2 Les protocoles d'agrégation dans un arbre

► COUGAR

Le protocole COUGAR est un exemple d'agrégation de données distribuée. Les données produites par le réseau sont modélisées comme une table relationnelle, le réseau étant vu comme une grande base de données distribuée. Les attributs de cette table sont soit des informations sur le capteur soit des données produites par ce capteur car COUGAR fournit une agrégation partielle au niveau des nœuds.

En effet, chaque nœud contient une liste d'attente de tous les nœuds fils qui doivent lui envoyer des données. Le nœud n'émet le paquet agrégé que lorsque tous les fils de sa file d'attente lui ont envoyé des données.

Ce protocole nécessite donc une grande synchronisation afin de traiter les données dans le réseau (il faut attendre toutes les données avant d'effectuer le calcul de l'agrégation) [23].

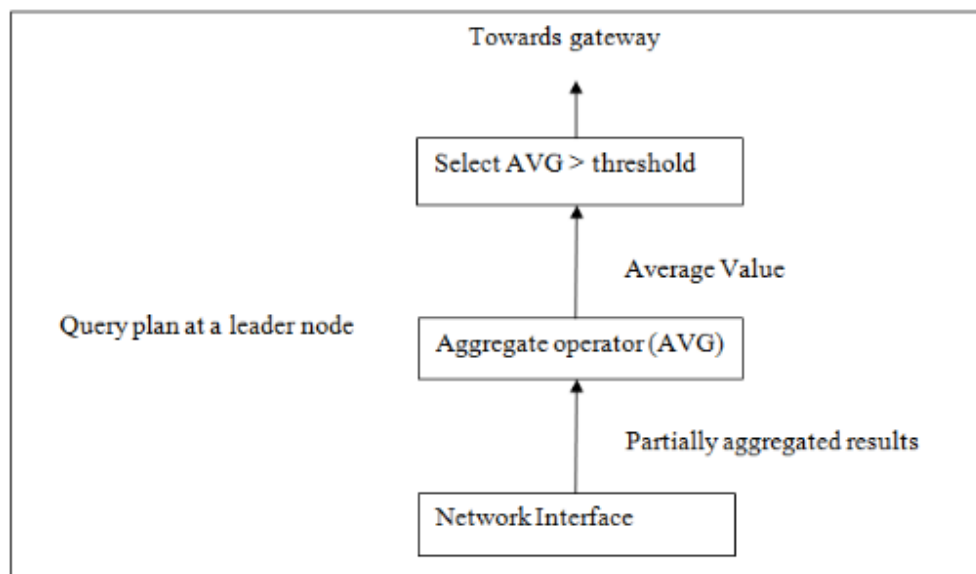


FIGURE 2.10 – Plan de requête au leader [23].

► TAG

Dans TAG, les résultats approximatifs de l'agrégation sont autorisés à améliorer l'efficacité énergétique et de réduire les coûts de communication. Des résultats approximatifs sont utiles pour la surveillance en ligne et peut prendre en charge les mécanismes de mise en réseau pour la récupération d'erreur en réseau. Et cette réponse partielle permet aux utilisateurs d'affiner dynamiquement leurs requêtes à l'aide d'agrégation en ligne [39].

► **Temporal coherency-aware in-Network Aggregation (TiNA)**

(TiNA) fonctionne au sommet d'un arbre de routage (c.-à-d TAG ou Cougar) ayant le point des données collecté (puits) en tant que racine. Elle exploite la corrélation temporelle dans une séquence de lectures du capteur afin de réduire la consommation d'énergie en supprimant les valeurs qui n'affectent pas la qualité attendue des données agrégées. Ceci est implémenté grâce à la clause de TOLERANCE qui est ajouté à la requête SQL [34].

2.4 Conclusion

Les réseaux de capteurs se caractérisent par une capacité énergétique limitée rendant l'optimisation de sa consommation dans ce type de réseaux une tâche critique pour prolonger sa durée de vie. Cependant, nous avons remarqué que plusieurs facteurs et contraintes compliquent la gestion de ce type de réseaux. C'est pour cela d'ailleurs que plusieurs travaux ont été établis dans la direction d'émetteur/récepteur cherchant à fournir de meilleurs gains énergétiques.

Dans ce deuxième chapitre, nous avons présenté les mécanismes de traitement de données tel que la compression qui est utilisé, tout en assurant une perte tolérable ou minimale d'information. L'agrégation de données afin d'obtenir des volumes plus petits à transférer, comme on a présenté le traitement des requêtes dans les réseaux de capteurs ainsi que la gestion de données collecté et l'application de l'agrégation dans les réseaux de capteurs.

Dans le chapitre suivant nous allons étudier Les vues matérialisées, et leurs rôles dans les réseaux de capteurs.

CHAPITRE 3

APPROCHES DE GESTION DE DONNÉES DANS LES RCSFS AVEC LES VUES MATÉRIALISÉES

3.1 Introduction

L'intérêt croissant pour le traitement des données créées par les réseaux de capteurs s'est développé pour implémenter les réseaux de capteurs comme des bases de données. Dans la phase de récolte, les données du capteur sont transmises au nœud puis (également désigné comme station de base ou nœud interrogeant) immédiatement après son acquisition du monde physique. Bien que l'agrégation en réseau réduise de manière significative la consommation d'énergie, la transmission inconsciente de tous les résultats des requêtes à partir de tous les capteurs à chaque tour d'acquisition est actuellement le facteur le plus exigeant d'énergie dans de tels environnements [36], [37], [32].

Dans ce chapitre, nous abordons une étude critique des approches exploitant des vues matérialisées dans les réseaux de capteurs généralement basé sur l'agrégation des données. Enfin, nous terminons par une comparaison des approches étudiées à base de leurs performances.

3.2 Définitions et concepts

Dans cette section, nous définissons quelques concepts qui seront utiles dans la description dans les différentes approches.

3.2.1 Vues

Une vue est une relation virtuelle au sens où ses instances n'existent pas physiquement mais elles sont calculées à chaque invocation de la vue. Une vue est définie par une requête qui utilise des relations ou des vues existant[38].

3.2.2 Vues matérialisées

Une vue matérialisée est une table contenant les résultats d'une requête. Les vues améliorent l'exécution des requêtes en pré calculant les opérations les plus coûteuses comme la jointure et l'agrégation, et en stockant leurs résultats dans la base. En conséquence, certaines requêtes nécessitent seulement l'accès aux vues matérialisées et sont ainsi exécutées plus rapidement [38].

Comme un cache, une vue matérialisée fournit l'accès rapide aux données, les vues matérialisées sont utiles dans les nouvelles applications telles que les données entreposant, elles sont utilisées pour extraire les données dans l'entrepôt. Ces vues sont dérivées à partir de table de la base source.

3.2.3 Maintenance des vues matérialisées

Les données de table sources peuvent évoluer dans le temps à cause des mises à jour, alors que les vues dérivées de ces tables restent inchangées, c'est à dire leurs données ne sont pas cohérentes avec celles des sources. Ce problème est appelé le problème de maintenance des vues matérialisées.

Trois stratégies fondamentales ont été proposées pour le problème de maintenance des vues matérialisées.

Dans la première stratégie, les vues sont mises à jour périodiquement.

Dans la deuxième stratégie, les vues sont mises à jour immédiatement à la fin de chaque transaction.

Dans la dernière stratégie, les modifications sont propagées d'une manière différée. Dans ce cas, une vue est mise à jour uniquement au moment où elle est utilisée par une requête d'un utilisateur " à la demande " [38].

3.2.4 Ensemble de limites k-couvert (V'_i)

Soit v un tuple, v^{ub} est sa borne supérieur (sa valeur maximale), V_i est la vue d'un certain nœud capteur arbitraire (S_i), V'_i est un sous ensemble des tuples appartenant à la vue V_i , et v_k^{lb} est la k^{eme} plus grande borne inférieur :

Si $v \notin V'_i$ alors $v^{ub} < v_k^{lb}$ [40].

3.2.5 Requêtes d'agrégation spatiale

La requête d'agrégation spatiale est exprimée en SQL :

SELECT AGG (lecture)

FROM Réseau de capteur

WHERE localisation IN ($[x1, x2], [y1, y2]$) ;

L'expression (AGG (lectures)), signifie la collecte des lectures agrégées des capteurs dans un réseau de capteurs, dont les localisations se situent dans une région déterminée par ($[x1, x2], [y1, y2]$) [41].

3.2.6 Fenêtre des requêtes

C'est la zone de localisation qui situe dans une région déterminée par ($[x1, x2], [y1, y2]$) Où $x1, x2, y1$ et $y2$ sont les coordonnées d'une grille [41].

3.2.7 Réseau de capteurs fédéré

Un réseau de capteurs fédéré se compose d'un ensemble de réseaux locaux interconnectés, de telle sorte que chaque nœud N_i maintient un ensemble de flux de données $N_j = s_1, s_2, s_b$, $j \in [1; a]$ dans un réseau de capteurs locale[42].

3.2.8 Nœud consommateur et nœud producteur

Dans un réseau de capteurs fédéré, il arrive souvent qu'un nœud a besoin d'apporter les flux de données maintenues par un autre nœud (ou plusieurs autres nœuds) à travers l'Internet, par l'analyse des données ou de traitement des requêtes. Le premier nœud est appelé un nœud consommateur et ce dernier nœud un nœud producteur. Il est également possible que le nœud producteur aurait besoin des flux de données gérées par le nœud consommateur, et donc les deux nœuds deviennent producteurs et consommateurs en même temps [42].

3.3 Approches de gestion de données dans les RCSFs avec les vues matérialisées

Nous allons dans ce qui suit présenter les approches proposées pour l'agrégation des données en utilisant les vues matérialisées dans les RCSFs :

3.3.1 L'approche de vue matérialisée pour maintenir les opérations d'agrégation à travers de longues périodes dans les réseaux de capteurs

Description de l'approche

Cet approche proposée par Minsoo Lee, Julee Choi, and Sookyung Song [27], propose d'ajouter la clause de DURING dans TinySQL pour calculer les valeurs pendant une période spécifique de temps et propose une façon de mettre en œuvre le service d'agrégation dans les réseaux de capteurs en utilisant les vues matérialisées et les techniques de la maintenance incrémentales de vues utilisées dans les entrepôts de données.

Les fonctions d'agrégation sont composées d'une fonction f pour la fusion, une fonction I pour l'initialisation et une fonction e pour l'évaluation.

Dans le système actuel de base de données du réseau de capteur, il n'est pas possible d'obtenir la valeur d'agrégation sur une longue période de temps parce que les capteurs n'ont pas la puissance ou de la mémoire pour accumuler les données détectées pendant une longue période de temps. Afin de pouvoir soutenir ce type d'agrégation, tout d'abord l'ajout d'une clause DURING à la syntaxe TinySQL est proposé.

Pour calculer la valeur d'agrégation basée sur la clause DURING, les techniques des vues matérialisées et la maintenance incrémentale des vues sont proposés d'être utilisés. Pour avoir les valeurs de la température moyenne de chaque groupe, on peut définir une vue matérialisée à l'aide de la TinySQL étendue et la matérialiser au niveau du nœud de station de base pour un meilleur traitement. A la réception de la requête et le prédicat (permet au nœud donné de le choisir son groupe), les nœuds fils choisissent leur propre groupe et retournent les données agrégées (id de groupe, des données détectées, count).

En utilisant cette méthode, au cours d'une longue durée, toutes les données détectées doivent être stockées. Par conséquent, si le délai est beaucoup plus long, le système et les nœuds capteurs connaîtront une charge supplémentaire. Ainsi, en considérant les limitations des nœuds capteurs, cette solution n'est pas suffisante.

En utilisant cette méthode, au cours d'une longue durée, toutes les données détectées doivent être stockées. Par conséquent, si le délai est beaucoup plus long, le système et les nœuds capteurs connaîtront une charge supplémentaire. Ainsi, en considérant les limitations des nœuds capteurs, cette solution n'est pas suffisante. Pour remédier à ce problème les auteurs suggèrent que l'approche de la maintenance incrémentale des vues soit prise en compte pour mettre à jour efficacement la vue matérialisée. L'instruction suivante montre la formule de maintenance incrémentale des vues . Les changements (S_i) sur les relations de base (S_i) peuvent être calculés et appliqués à la vue en utilisant la formule ci-dessus[27].

$$\begin{array}{l}
 \Delta V = (\Delta S_1 \bowtie S_2 \bowtie S_3 \dots \bowtie S_n) \\
 \cup \\
 (S_1' \bowtie \Delta S_2 \bowtie S_3 \dots \bowtie S_n) \cup \\
 \dots \\
 (S_1' \bowtie S_2' \bowtie S_3' \dots \bowtie \Delta S_n) \\
 V' = V \cup \Delta V
 \end{array}$$

Cette formule représente donc le changement (ΔV) de la vue V , qui peut être effectué par ΔS_1 de la relation S_1 , en suite une jointure entre ΔS_1 et toutes les autres relations (S_2, S_3, \dots, S_n) doit être effectuée, et de même pour chaque changement de chaque relation avec toutes les autres relations, au finale l'union de ces changements est calculée puis appliquée à la vue.

Le tableau 3.1 montre la formule utilisée pour plusieurs opérateurs d'agrégation pour la maintenance incrémentale des vues. $\Delta + (S)$ est la valeur d'insertion à appliquer à la vue, ou en d'autres termes le résultat de détection de la requête à chaque époque et MAX' est la valeur MAX recalculée.

MAX	if $\Delta + (S) > MAX$ then $MAX' = \Delta + (S)$
MIN	if $\Delta + (S) < MIN$ then $MIN' = \Delta + (S)$
SUM	$SUM' = SUM + \Delta + (S)$
COUNT	$COUNT' = COUNT + \Delta + (S)$
AVG	$AVG' = (AVG * COUNT + \Delta + (SUM)) / COUNT + \Delta + (COUNT)$

TABLE 3.1 – les formules de la maintenance des vues incrémentales.

Avantage

– La maintenance incrémentale des vues matérialisées permet de calculer uniquement les modifications qui doivent être appliquées à la vue, ainsi réduire la taille et le nombre de messages.

Inconvénient

– L'utilisation de cette approche peut ne pas être efficace dans quelques situations. Par exemple, si la requête est de calculer la température moyenne pendant 3 minutes, en utilisant des vues matérialisées n'est pas efficace parce que le nœud de capteur peut avoir assez de mémoire et n'a pas besoin de prendre en plus les couts de mise à jour des vues matérialisées.

3.3.2 Les vues MINT : Les vues matérialisées top-k distribuées

Description de l'approche

Les auteurs de [40] ont proposé l'approche des vues MINT (Materialized In-Network Top-k), permet de modéliser la récupération de données sur la supposition que l'utilisateur n'est intéressé que par les k réponses les mieux classés plutôt que toutes les réponses, l'objectif est d'optimiser les requêtes top-k sur plusieurs tuples de réponses c'est un nouvel outil pour réduire le nombre de messages et la consommation d'énergie ainsi le temps d'exécution des requêtes de surveillance continue.

Les algorithmes sous-jacents de l'approche de Vue MINT supportent également, les Vues INT (In-network Top-k), la version de MINT apatriide qui est approprié aux capteurs de mémoire limitée. L'approche MINT fonctionne en trois phases : La phase de création, La phase

d'élagage et la phase de mise à jour.

1. La phase de création

Cette phase est exécutée lors de la première acquisition de lectures qui se traduit par (n) vues distribuées, V_i ($i \leq n$). Chaque capteur exécute localement Q , il agrège les résultats avec les réponses de requêtes du nœud fils, Cela génère une vue locale V_i .

2. La phase d'élagage

Cette phase illustre l'élagage de la vue (V_i) d'un certain nœud capteur arbitraire (S_i) et la construction de l'ensemble des tuples V'_i appartenant à la vue V_i . Cet algorithme est appliqué à la fois aux approches MINT et INT. La première étape identifie le seuil d'élagage noté v_k^{lb} (la k^{eme} plus haute borne inférieure). Ce seuil permet à l'algorithme d'élaguer les tuples qui ne seront pas retenus dans le résultat V'_i . La prochaine étape consiste à localiser les tuples qui ont une borne supérieure v^{ub} en dessous du seuil v_k^{lb} , un moyen efficace de le faire est de créer une liste ordonnée de limites supérieures et ensuite effectuer une analyse linéaire par ordre décroissant jusqu'à où se trouve un tuple v_j^{ub} ($< v_k^{lb}$). Tout tuple de borne supérieure de valeur $\leq v_j^{ub}$ peut être éliminé. Finalement une analyse linéaire dans l'ordre décroissant est effectuée et elle s'arrête lorsqu'un tuple v^{ub_j} est inférieur à v_k^{lb} .

3. La phase de mise à jour de MINT

Dans l'étape précédente, la vue V_i est transformée à un sous-ensemble V'_i élagué. Afin de minimiser la communication, seulement le nœud fils(S_i) retransmet la vue V'_i à son nœud père, si la vue V'_i a changé. Toute mise à jour d'un tuple x avec une borne supérieure (notée x^{ub}) de valeur inférieure au seuil v_k^{lb} peut être ignoré. Dans le cas contraire, nous ajoutons le tuple x pour l'ensemble de tuples candidats dans V'_i . Si $x^{ub} \leq v_k^{lb}$ (i.e. la la valeur minimale du tuple x est inférieure au seuil v_k^{lb}) est vrai, alors le seuil v_k^{lb} n'a pas changé. Par conséquent, le nœud (S_i) seulement propage la mise à jour x vers son parent plutôt que de recalculer la vue complète. Dans la mise en œuvre ils mettent en tampon ces mises à jour jusqu'à ce que tous les nœuds fils envoient leurs mises à jour à leurs nœuds pères. Si dans le cas contraire le seuil $v_k^{lb} < x^{ub}$ (le seuil v_k^{lb} est inférieur à la valeur minimale du tuple x), alors le seuil v_k^{lb} changera. En conséquence le nœud (S_i) doit reconstruire la vue V'_i et transmettre complètement la vue V'_i à son parent. La reconstruction ne se produit que pour $|V'_i|$ éléments (le nombre d'élément dans V'_i) plutôt que tous les éléments (i.e, $|V_i|$).

Avantage

- MINT propose des réductions d'énergie considérables dans l'exécution de requêtes de surveillance continue.
- Réponse plus rapide aux utilisateurs par l'optimisation de requêtes top-k .

Inconvénient

- Lorsque les réponses exactes ne sont pas nécessaires, la solution n'est pas efficace pour les vues approximatives top-k.
- L'élagage nous conduit à la perte de quelque information détectées, elle ne garde pas la totalité des détections.

3.3.3 Traitement des requêtes d'agrégation multiples dans les réseaux Geo-capteur

Les auteurs [41] ont proposé des approches permettant de traiter les requêtes d'agrégation émis par les différents capteurs formant ainsi des points d'accès dans ce réseau, en examinant des stratégies pour maintenir les vues matérialisé dans le réseau (MINVs : Materialized In-Network Views) qui pré-calculent et stockent les résultats d'agrégation couramment utilisés dans le réseau de capteurs. Avec MINVs, les résultats détectés et agrégés pour certaines régions géographiques sont disponibles et partageable entre les requêtes. Ainsi, le nombre d'accès de capteur est considérablement réduit.

Les auteurs ont proposé trois approches différentes pour la gestion d'une MINV et le traitement des requêtes d'agrégation spatiale, à savoir :

3.3.3.1 Analyse complète

L'analyse complète est une approche de traction pure. Elle maintient aucune vue. Chaque requête doit parcourir tous les capteurs à l'intérieur de la fenêtre de requêtes et collecte et agrège les lectures. Initialement, une requête est acheminé à partir du point d'accès le plus proche du capteur, nommé le nœud originaire, sur la limite de la fenêtre spatiale. Le parcours de requête suit deux sortes de chemins, à savoir, le chemin frontière et les chemins intérieurs. Pour une seule requête, l'analyse complète prend un seul passage et visite une fois tous les capteurs. Cependant, les capteurs dans une région commune peuvent être consultés à plusieurs reprises par les requêtes.

3.3.3.2 Avantages

- Les chemins assurent une couverture complète de tous les capteurs dans la fenêtre.
- Le parallélisme du chemin frontière et des chemins intérieurs améliore le temps de réponse.
- Le nœud d'origine dans lequel l'analyse termine est le plus proche du capteur du point d'accès ; facilite la livraison efficace du résultat final.

3.3.3.3 Inconvénient

- L'analyse complète peut soutenir tous les types des fonctions d'agrégation, mais elle est généralement moins performante que les deux autres régimes en raison du coût élevé du parcours.

3.3.3.4 Réplication du Cluster

L'approche comme son nom l'indique maintient une vue d'un groupe prédéfini et reproduit la vue à tous les nœuds à l'intérieur du cluster.

Pour maintenir la fraîcheur d'une vue à l'intérieur d'un cluster, chaque capteur connaît d'autres capteurs membres dans le même groupe et qu'ils adoptent un mécanisme d'inondation de mettre à jour la vue. Quand un capteur obtient une nouvelle lecture, il met à jour sa propre réplique de vue et diffuse ensuite le changement (la différence entre les lectures précédentes et la nouvelle somme sémantique) pour les nœuds voisins intermédiaires. L'émission est marquée par un tuple unique (ID_s, ID_b) , Avec ID_s l'ID de capteur émetteur et ID_b l'ID de la diffusion de l'expéditeur. Une fois que les nœuds voisins reçoivent les modifications, ils mettent à jour leurs propre vue réplique et retransmettre le même message.

Avantage

- La traversée de tous les nœuds membres d'un cluster est remplacée par un capteur de visite, ce qui permet de réduire de manière significative le nombre d'accès de capteurs.

Inconvénient

- La taille de cluster a un impact sur la performance.
- Le coût est important lors de la mise à jour de la vue
- Le gaspillage mémoire

3.3.3.5 Somme préfixe

La troisième approche est la vue de somme préfixe qui accumule des lectures dans tout le réseau. Chaque capteur maintient une somme cumulée de lectures partielles. Ils présentent les

deux variantes, à savoir :

la somme préfixe à une dimension (1-D)

Avec la somme préfixe 1-D, chaque capteur, $s_{i,j}$ dans un réseau de capteurs maintient sa propre lecture $r(s_{i,j})$, et une somme cumulée des lectures indiquées par : $V_{i,j-1}$ (ou simplement $V_{i,j-1} = r(s_{i,j-1}) + V_{i,j-2}$, on suppose que $V_{i,-1} = 0$). Le capteur $s_{i,j}$ connaît la somme des lectures de tous les nœuds précédents notamment $r(s_{i,0})$, $r(s_{i,1}) \cdot \dots \cdot r(s_{i,j-1})$ de la ligne i .

A la ligne i , le premier nœud, $S_{i,0}$ commence à propager ses lectures en tant que $V_{i,0}$ ($r = (S_{i,0})$) vers le second nœud, $s_{i,1}$. Ensuite, $s_{i,1}$ maintient cette lecture reçue, et calcule $V_{i,1}$, c'est-à-dire $V_{i,0} + r(s_{i,1})$ et l'envoie à $S_{i,2}$. Les propagations se répètent jusqu'à ce que le dernier capteur de la rangée soit atteint.

Sur demande, un capteur $s_{i,j}$ peut rapporter n'importe laquelle de ces trois lectures : (1) la lecture, $r(s_{i,j})$, (2) $V_{i,j-1}$, une somme cumulée stockée de $r(s_{i,0})$ par l'intermédiaire $r(s_{i,j-1})$ et (3) $V_{i,j}$, une somme cumulée dérivée à partir des lectures de $S_{i,0}$ à $S_{i,j}$ (soit $V_{i,j-1} + r(S_{i,j})$).

Sur demande, un capteur $s_{i,j}$ peut rapporter n'importe laquelle de ces trois lectures : (1) la lecture, $r(s_{i,j})$, (2) $V_{i,j-1}$, une somme cumulée stockée de $r(s_{i,0})$ par l'intermédiaire $r(s_{i,j-1})$ et (3) $V_{i,j}$, une somme cumulée dérivée à partir des lectures de $S_{i,0}$ à $S_{i,j}$ (soit $V_{i,j-1} + r(S_{i,j})$).

la somme préfixe à deux dimensions (2-D)

Est dans un mode à 2 dimensions. La notation $W_{i,j}$ est la somme cumulée de la lecture de $S_{0,0}$ à $s_{i,j}$. Chaque capteur $s_{i,j}$ doit maintenir quatre lectures, y compris (1) la lecture locale, $r(s_{i,j})$, (2) la somme cumulée $W_{i,j-1}$ fourni par nœud $s_{i,j-1}$, (3) la somme cumulée $W_{i,j}$ fourni par le nœud $s_{i-1,j}$, et (4) la somme cumulée $W_{i-1,j-1}$ à partir du nœud $s_{i-1,j-1}$. Avec ces quatre lectures, $s_{i,j}$ peut déterminer $W_{i,j}$ qui est $r(s_{i,j}) + W_{i,j-1} + W_{i-1,j} - W_{i-1,j-1}$.

Quand un capteur $s_{i,j}$ reçoit toutes les sommes cumulées de ses voisins, il calcule $W_{i,j}$ et diffuse deux lectures, $W_{i,j}$ et $W_{i,j-1}$ pour les capteurs $s_{i,j+1}$ et $s_{i+1,j}$.

Les lectures agrégées pour une requête peuvent être obtenues de manière efficace avec

moins de visites de capteurs. Pour une région $[x1, x2]$, $[Y1, Y2]$, la lecture agrégée est $W_{x2,y2} - W_{x2,y1-1} - W_{x1-1,y2} + W_{x1-1,y1-1}$.

Avantages

– La somme préfixe offre une meilleure réutilisation d'une MINV par l'addition et la soustraction simple. La vue de somme préfixe 1-D à un coût de maintenance légèrement inférieure et un accès concurrentiel plus élevé dans la mise à jour de vue que la vue de somme préfixe 2-D. La surcharge de stockage pour une vue de somme préfixe 1-D est inférieure à une vue de somme préfixe 2-D.

3.3.4 Approche basée sur les modèles

Description de l'approche

A. Arion, H. Jeung, K. Aberer et S. Michel [42] proposent une nouvelle approche qui emploie les avantages de l'utilisation des vues basées sur des modèles (fonctions qui reflète la variation des valeurs détectées dans le temps) pour une communication efficace des flux de données traités sur les réseaux de capteurs fédérées. Soit $s = \langle v_1, v_2, \dots, v_n \rangle$ un flux de données brutes récoltées à partir d'un capteur, représenté par une séquence de lectures estampillées, où $v_t \in S$ indique la valeur à l'instant t . La transmission de flux de données brutes sur un réseau fédéré nécessite des coûts de communication de données élevés. Pour remédier à ce problème, les auteurs utilisent des vues basées sur des modèles pour un traitement efficace de flux de données distribuées. Cette approche se déroule en deux phases :

- **Phase d'initialisation** : quand un nœud consommateur demande un flux de données à un nœud producteur, le nœud producteur construit un modèle (temps paramétrable) qui peut prédire les valeurs aux moments actuels ou futurs. Le nœud producteur envoie alors les paramètres du modèle pour le consommateur et le nœud consommateur construit le même modèle en utilisant les paramètres reçus. Cette opération est effectuée une seule fois lorsque la connexion entre les deux nœuds est initialement établie.
- **Phase d'exécution** : le nœud consommateur génère une vue basée sur un modèle comme le flux brut est représenté par le modèle construit. Toute requête au niveau du nœud consommateur est directement traitée sur la vue à base de modèle sans aucune communication de données avec le nœud producteur. L'approche actualise ensuite les paramètres du modèle à partir du nœud producteur au nœud consommateur uniquement

lorsque c'est nécessaire.

Lorsque l'approche construit un modèle à un nœud producteur, elle prend une erreur spécifiée par l'utilisateur à destination, telle que la différence entre une mesure de capteur brut et la valeur correspondante dans une vue à base de modèle, appelé la valeur guidée par le modèle, ne doit pas dépasser une limite. L'approche fournit cette garantie d'exactitude dans tout le réseau de capteurs fédéré, formellement déclaré dans la propriété suivante :

Propriété :

Soit $s = \langle v_1, v_2, \dots, v_n \rangle$ un flux de données brutes et $s' = \langle v'_1, v'_2, \dots, v'_n \rangle$ une vue à base de modèle créé par le modèle Ms. Étant donné une exactitude limitée ϵ_s pour s , l'approche garantit que :

$$|v_t - v'_t| \leq \epsilon_s \text{ tel que } v_t \in s, v'_t \in s'.$$

Afin de maintenir la propriété, l'approche réalise la mise à jour du modèle d'un nœud producteur aux nœuds consommation selon les étapes suivantes :

- Le nœud producteur génère une valeur guidée par le modèle quand une nouvelle lecture brute est transférée, et vérifie si la différence entre la valeur brute et la valeur guidée par le modèle reste dans la limite d'erreur.
- Si la différence ne dépasse pas la limite d'erreur, aucune communication n'est nécessaire entre les deux nœuds, et le nœud consommateur génère des valeurs pour leurs vues à base de modèle.
- Dans le cas contraire, le nœud producteur reconstruit son modèle, de sorte que la valeur guidée par le modèle générée à partir du modèle reconstruit ne dépasse pas la limite d'erreur par rapport à la lecture brut courante. Ensuite, le nœud producteur met à jour les modèles au niveau des nœuds consommateurs par l'envoi des nouvelles valeurs des paramètres du modèle reconstruit.

Bien que la transmission des valeurs de paramètres pour les modèles sur les réseaux soit beaucoup plus efficace que d'envoyer des flux de données réelles, cet avantage peut diminuer lorsque les valeurs dans le flux varient considérablement au cours d'un court terme et donc les mises à jour du modèle d'un nœud à l'autre se produisent souvent. Pour faire face à ce problème, un nouveau système qui permet de mettre à jour le modèle de manière plus efficace est introduit.

L'idée consiste à partager un ensemble de valeurs de paramètres prédéterminés pour la construction d'un modèle entre un nœud producteur et d'un nœud consommateur. L'approche transfère alors uniquement les identifiants bitmap-codés des paramètres préétablis lorsqu'une mise à jour de modèle est nécessaire, au lieu d'envoyer les valeurs réelles des paramètres pour le modèle.

Tout d'abord, le nœud producteur calcule deux ensembles de valeurs de paramètres prédéterminés pour le modèle linéaire (i.e. pentes et ordonnées v), de sorte que la valeur de chaque paramètre a un identifiant distinct. Il envoie ensuite les informations des valeurs prédéterminées pour le nœud consommateur lorsque la connexion entre les deux nœuds est d'abord établie. Par la suite, l'information est mise à jour uniquement lorsque cela est nécessaire.

Après l'initialisation, le nœud producteur représente le flux en tant que instance du modèle M_1 pendant $t \in [1, t_n]$, et il correspond à la fois les paramètres réels de M_1 aux valeurs des paramètres prédéfinis les plus similaires. Bien que cela, le nœud producteur surveille également si la propriété 1 détient entre chaque lecture réelle d'un capteur et sa correspondante valeur guidée par le modèle obtenu à partir du modèle prédéfini. Au niveau du nœud consommateur, la vue basée sur le modèle est générée par le modèle prédéterminé au cours de la même période de temps $t \in [1, t_n]$.

La lecture réelle à l'instant t_{n+1} au niveau du nœud producteur est supposé dépasser une limite d'erreur donnée à partir de sa correspondante produite par le modèle prédéfini, nécessitant le nœud producteur de reconstruire le modèle réel en tant M_2 . Le nœud producteur trouve alors les valeurs des paramètres prédéfinis les plus semblables à M_2 à nouveau. Ensuite, il encode un bitmap en utilisant les identifiants des valeurs des paramètres prédéfinis et envoie le résultat bitmap pour le nœud consommateur où se dérive les identificateurs de paramètres de décodage du bitmap. Le nœud de consommateur génère sa vue basée sur un modèle par le modèle construit par les valeurs de paramètres prédéterminés ayant leurs identificateurs.

Avantages

- Minimiser le coût de la communication en transmettant les valeurs des paramètres des modèles, au lieu d'envoyer des flux de données d'origine.
- Augmentation de la performance de l'approche par la mise à jour du modèle codé qui permet une mise à jour efficace du modèle d'un nœud à un autre.

Inconvénients

- Cet avantage d'efficacité peut diminuer lorsque les valeurs dans le flux varient considéra-

blement au cours d'un court terme et donc les mises à jour du modèle d'un nœud à l'autre se produisent souvent.

3.4 Analyse et comparaison

Les techniques de maintenance des vues pour l'agrégation de données peuvent procurer un gain intéressant en énergie dans le domaine des réseaux de capteurs, tel que montré dans les approches présentées précédemment, la différence réside dans l'utilisation des vues matérialisées et leurs maintenances, les approche MINT, MINVs et celle basée sur les modèles matérialisent des vues à travers le réseau, et chaque capteur effectue une mise à jour, par contre dans l'approche de vue matérialisée pour maintenir les opérations d'agrégation à travers de longues périodes dans les réseaux de capteurs la vue est créée au niveau du nœud de station de base pour le calcul efficace.

L'approche MINT et celle basée sur les modèles proposent de minimiser le coût de la communication donc des réductions d'énergie considérables, l'approche de vue matérialisée pour maintenir les opérations d'agrégation à travers de longues périodes permet de réduire la consommation de mémoire et de soutenir le calcul rapide et MINVs permet la réduction du temps de réponse aux requêtes.

Nous avons mis en évidence la technique l'approche de vue matérialisée pour maintenir les opérations d'agrégation à travers de longues périodes dans les réseaux de capteurs, étant plus efficace pour la réduction des coûts de mise à jour pour une longue durée d'agrégation.

	Mémoire	Energie	Nombre de message
Vue matérialisée pour maintenir les opérations d'agrégation à travers de longues périodes	Réduction de la consommation mémoire du capteur grâce à la mise à jour qui se fait en calculant uniquement les modifications qui doivent être appliquées à la vue.	Réduction de la consommation d'énergie.	L'approche permet de réduire le nombre de messages transmis, en exécutant les requêtes localement par la vue stocke dans le puits.
Les vues MINT	MINT est approprié aux capteurs avec une mémoire limitée.	Minimiser la consommation d'énergie en évitant la réexécution de la phase d'élagage pendant la phase de mise à jour.	Minimiser le nombre de messages entre les capteurs dans la phase d'élagage. Mais elle ne garde pas la totalité des informations détectées.
Traitement des requêtes d'agrégation multiples dans les réseaux Geo-capteur	Occupe un espace mémoire important dans un capteur.	Minimiser la consommation d'énergie par le nombre minimal de capteurs à visiter pour une requête.	L'approche permet de réduire le nombre de messages transmis.

Approche Basée sur les modèles	Une consommation considérable de l'espace mémoire du à la sauvegarde des valeurs des paramètres prédéterminés.	Suite aux mises à jour proposées le nombre de transmissions augmente, par conséquent la consommation d'énergie est élevée.	Bien que la taille des messages soit réduite, leur nombre augmente lorsque les mises à jour du modèle d'un nœud à l'autre se produisent souvent due à la variation considérable des valeurs dans le flux pendant un court terme.
---------------------------------------	--	--	--

TABLE 3.2 – Tableau de comparaison entre les quatre approches

3.5 Conclusion

Dans ce présent chapitre, nous avons présenté quelques techniques d'agrégation de gestion de données en utilisant les vues matérialisées, telle que l'approche de vue matérialisée pour maintenir les opérations d'agrégation à travers de longues périodes dans les réseaux de capteurs, MINT (Materialized In-Network Top-k Views in Sensor Networks), MINVs (Materialized In-Network Views), ces techniques peuvent réduire considérablement le temps de réponse aux requêtes et le nombre de messages ainsi procurer un gain intéressant en énergie dans le domaine des réseaux de capteurs, par contre la maintenance de ses vues engendre un coût supplémentaire.

CHAPITRE 4

LA PROPOSITION

4.1 Introduction

Les capteurs sont conçus pour fonctionner durant des mois et des années. Ainsi, la capacité énergétique des capteurs doit être utilisée efficacement afin de maximiser la durée de vie du réseau. A noter qu'une fois qu'un nœud capteur a épuisé son énergie, il est considéré comme défaillant. Ainsi, le développement d'une technique efficace permettant d'économiser la ressource énergétique est un objectif primordial pour les réseaux de capteurs sans fil.

Dans ce chapitre, nous proposons une approche distribuée, basée sur l'idée de ne transmettre les données captées que si elles changent de classes. Ce qui permettra de réduire donc la redondance des données, nous avons besoin de déterminer quelles sont les hypothèses de travail que nous allons préalablement poser.

4.2 Hypothèses et exigences

Afin de déterminer le travail que nous avons à faire, il est important de fixer le cadre dans lequel nous allons évoluer. Nous avons donc posé certaines hypothèses de travail.

4.2.1 Hypothèses de travail

H1. Nous supposons que l'ensemble des nœuds du réseau sont adressables, c'est-à-dire qu'on pourra établir une communication directe entre deux nœuds.

H2. Les capteurs ont quatre fonctions de base :

- La détection : Les capteurs peuvent détecter et collecter les lectures
- Le stockage : servir de stockage pour ses propres lectures recueillies et d'une vue matérialisée.
- Le calcul : permet quelques tâches de traitement de requête
- La communication sans fil : permet à un capteur de relayer les messages d'un capteur à ses voisins.

H3. Tous les capteurs sont considérées identiques en termes de capacité de traitement, capacité de la mémoire, couverture de transmission et d'énergie.

H4. Nous supposons de plus que les capteurs sont conscient de l'emplacement (par exemple, chaque capteur obtient sa situation géographique) et le temps synchronisé, (c'est à dire, chaque capteur obtient l'horloge globale) par GPS ou d'autres techniques de positionnement.

4.2.2 Exigences

À partir des hypothèses que nous avons formulées au point précédent, nous allons ici définir une série d'exigences auxquelles notre architecture devra répondre.

D'un point de vue plus général, nous devons assurer la fiabilité du réseau. Nous devons notamment pouvoir ajouter ou retirer des nœuds du réseau à tout moment. Nous devons aussi assurer une optimisation de la charge sur le réseau de manière à limiter la consommation des ressources et par conséquent prolonger la durée de vie du réseau.

Dans le cas du système de détection d'incendie, par exemple, il pourrait être utile de conserver quelques valeurs passées des capteurs, permettant ainsi de remonter facilement à la source du feu. De manière plus pratique, stocker dans une vue matérialisée les valeurs captées par les nœuds, pourrait également permettre de limiter l'utilisation des ressources des capteurs.

En se basant sur ces hypothèses et exigences, nous proposons donc une nouvelle solution utilisant la notion de vue matérialisée et la technique d'agrégation des données pour réduire le nombre de transmission dans un réseau de capteurs sans fil. Nous commençons par sa description.

4.3 Présentation du système

Dans cette partie nous allons présenter une architecture répondant à l'ensemble des exigences que nous avons présentées au point 4.2.

4.3.1 Les différents types de nœuds du système

Nous avons introduit dans l'architecture trois entités différentes. Chaque entité définit des capacités très précises. Certaines entités sont communes à toutes les architectures de réseau de capteurs sans fils. Voici une liste exhaustive des compétences de chaque entité.

4.3.1.1 Le capteur (*sensor*)

Le rôle de capteur, comme le dit bien son nom, définit la capacité d'un nœud à effectuer des mesures pour ensuite les envoyer soit à un agrégateur, soit à un utilisateur final en réponse à une requête.

4.3.1.2 L'agrégateur (*aggregator*)

Les agrégateurs vont être en charge de sauvegarder les valeurs reçues de plusieurs capteurs grâce à une vue matérialisée et de les agréger pour ensuite sauvegarder le résultat dans la vue puis l'envoyer au nœud puits.

4.3.1.3 Le puits

La capacité d'un nœud à agir en tant que puits définit le fait qu'il va faire tourner l'application qui va avoir à utiliser les données issues du réseau de capteurs. Cela implique donc la capacité à envoyer des requêtes directement à un capteur.

4.4 Définition des symboles

Nous définissons dans tableau l'ensemble des symboles utilisés dans le détail de l'approche proposée :

Symboles	Définition
S_i	Le nœud capteur
V_i	La vue matérialisée stocké dans le capteur.
Q	La requête déclenchée par le puits.
Id	L'identifiant d'un nœud capteur (présenter par l'indice de capteur S_i) stockée dans la vue.
Val	Variable stockée dans la vue, contient la valeur captée (dans les nœuds extrêmes) et la valeur agrégée (dans les nœuds agrégateurs).
Val_1	Variable stockée dans la vue au niveau d'un nœud agrégateur, elle stocke les valeurs reçues de la part des nœuds fils.
Id_1	L'identifiant du nœud capteur fils au, stocké dans la vue au niveau du nœud agrégateur.
T_i	L'instant de la récolte des données.

TABLE 4.1 – Définition des symboles

4.5 Présentation générale de l'approche de segmentation de données

Notre solution se base sur la segmentation des données captées en classes et de transmettre les nouvelles données que dans le cas de changement de classe. Donc, l'objectif de l'approche de segmentation de données est de réduire la quantité de données (le nombre de messages) à transmettre sur le réseau et du coup réduire l'énergie consommée à cet effet.

L'approche de segmentation de données utilise des vues matérialisées au niveau de chaque capteurs, pour y stocker les données de ses fils ainsi que ses propre données captées. Et un

buffer pour stocker les classes de données déjà définies.

Notre solution suppose également qu'il n'est pas nécessaire de transmettre les données captées sauf si celle-ci n'appartient pas à la classe de l'ancienne valeur déjà captée. Cela permet de réduire le nombre de données à transmettre sur le RCSF. Pour ce faire, une vue matérialisée est définie au niveau d'un nœud. Cette vue contient la dernière valeur captée par les nœuds. A chaque capture, la valeur captée est vérifiée si elle appartient à la classe de données de la variable, si elle n'appartient pas, la valeur fera partie des données à transmettre, sinon elle est simplement pas envoyée.

Les données seront transmises aux agrégateurs qui appliqueront la fonction d'agrégation et les résultats seront envoyés au nœud puits qui appliquera à son tour la fonction d'agrégation aux données, et le résultat sera transmis à la station de base qui va prendre sa décision.

Les paragraphes qui suivent décrivent la solution proposée de manière plus détaillée.

4.6 Description détaillée de l'approche de segmentation de données

4.6.1 La phase de pré-exécution

– Segmentation des données (Définir les classes)

Chaque nœud capteur contient une liste de classes de données à traiter, chaque classe est définie par ses deux bornes minimale et maximale. Ainsi à partir des données récoltées : $n_1 \dots n_m$ nous allons créer les classes $C_1 \dots C_t$ de la forme : $C_1 [val_{min1} \ val_{max1}] C_t [val_{mint} \ val_{maxt}]$. Chaque classe reflète une situation et une décision différente des autres classes.

– Création des vues

Chaque nœud capteur maintient une vue matérialisée permettant de stocker les dernières données envoyées par ses nœuds fils pendant cette phase et sa propre valeur captée par son unité de capture.

4.6.2 La phase d'exécution

– La phase d'initialisation

La phase d'initialisation est lancée par le puits, chaque nœud qui reçoit la requête la propage vers ses nœuds fils, et procède à la capture d'une donnée à son niveau. Comme c'est la première collecte, chaque nœud i insère la valeur captée dans la vue, cette valeur sera envoyée au nœud de niveau inférieur (agrégateur) qui appliquera la fonction d'agrégation et ainsi de suite jusqu'au nœud puits.

– La phase de collecte de données

La collecte des données se fait au niveau des nœuds capteurs. Nous supposons que cette phase est déclenchée à la demande au niveau du puits qui transmet une requête de collecte Q sur le RCSF.

A l'arrivée de l'événement déclencheur de cette phase (une requête de collecte Q), le nœud capteur transmet à son tour la requête de collecte

1. Au niveau d'un capteur extrême

Le capteur sollicite son unité de capture, il reçoit la valeur captée. Celle-ci est sauvegardée dans la vue en remplaçant la valeur déjà sauvegardée (ou sauvegardée avec les valeurs précédentes en cas où les capteurs disposent d'un espace mémoire suffisant), et cette nouvelle valeur vérifie si elle appartient à la même classe que la dernière valeur stockée dans la vue, si c'est le cas alors cette valeur ne sera pas envoyée au nœud parent, sinon la valeur captée sera envoyée au nœud agrégateur.

2. Au niveau d'un agrégateur

Nous distinguons deux types de collecte des données au niveau de chaque capteur agrégateur : la collecte des données venant de son unité de captage et celles en provenance des nœuds capteurs qui sont ses voisins (dans sa portée). À chaque réception d'un paquet provenant de l'un de ses voisins, l'agrégateur stocke les données extraites du paquet, et applique la fonction d'agrégation aux données reçues, et la valeur obtenue (valeur agrégée) vérifie si elle appartient à la même classe que la dernière valeur agrégée, si c'est le cas alors il n'envoie pas le résultat, sinon la nouvelle valeur agrégée sera envoyée au nœud puits.

3. Au niveau du nœud puits

La phase de collecte des données au niveau du nœud puits est de même que la collecte des données au niveau du nœud capteur agrégateur à la différence que le nœud puits est connecté à l'ordinateur central de l'utilisateur final.

En effet, il serait important, pour un utilisateur final, de garder une trace de cette collecte de telle sorte qu'il puisse avoir l'information sur la donnée collectée à chaque moment et par quels nœuds sources ces données ont été collectées. Cela permettra à l'utilisateur de prendre quelques mesures d'intervention. Par exemple soit à une défaillance importante d'un certain nombre de nœuds capteurs, il pourra procéder à un redéploiement de d'autres capteurs.

La figure suivante est un organigramme résumant la phase de collecte de données au niveau des nœuds capteurs.

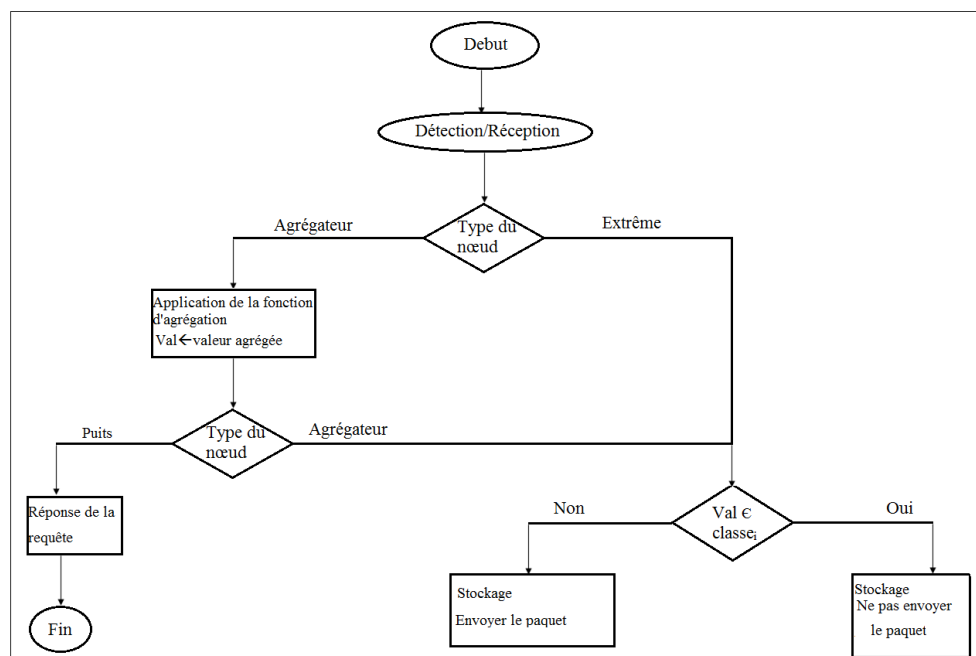


FIGURE 4.1 – Organigramme de la phase de collecte de données

4.7 Amélioration de l'approche proposée

L'objectif de cette amélioration est d'exploiter les capteurs hétérogènes et pour donner plus de précision aux données envoyées.

4.7.1 Le principe de l'amélioration

La phase de pré-exécution

- Chaque capteur contient des tableaux de classes de données à traiter, le nombre des tableaux de classes dépend de nombre de type de données à observer.
- Chaque nœud capteur contient des vues matérialisées pour stocké les dernière données récolter de chaque type de données.

La phase d'exécution

La phase de collecte de données suit le même algorithme de l'approche de segmentation de données :

- Quand un nœud puits déclenche une requête de collecte de données de l'un des types de donnée on les envoie suivant l'algorithme de l'approche1,
- A un moment donné une requête Q est déclenché par le nœuds puits qui demande la collecte des données d'un autre type suivant l'algorithme proposer dans l'approche1 et là on profite des envoie des données pour envoyer avec, les données du premier type qui ne sont pas envoyer a cause de la condition que ces données appartiennent au classes des dernières données envoyer ,le but de l'envoie de ces données est de donner plus de précision aux données envoyé au nœud puits et d'approcher plus au résultat final.

4.8 Déroulement de l'algorithme de l'approche de segmentation de données

Dans ce paragraphe, nous allons dérouler l'algorithme de l'approche de segmentation de données plus haut. Pour cela nous définissons le contexte dans lequel le déroulement sera effectué.

4.8.1 Présentation du scénario

Le domaine d'application utilisée et le domaine environnemental. Nous supposons que le type de données à observé et la température.



FIGURE 4.2 – Architecture générale de notre domaine d’application.

Nous supposons que le RCSF est constitué de dix (10) nœuds dont un est le puits. Les identifiants des nœuds sont sous la forme S_i , i allant de 1 à 10, S_{10} est l’identifiant du puits.

Pour mettre en évidence la réduction de la quantité de données transmises sur le RCSF et l’apport de l’algorithme en termes d’économie d’énergie, nous allons dérouler la collecte de données deux fois. Une première fois, la collecte se fait sans l’utilisation des classes de données (la phase d’initialisation). Une deuxième fois, la collecte se déroule selon l’algorithme proposé (phase de collecte de données), et les résultats seront représentés dans des tableaux.

4.8.2 Déroulement de l’exemple sans l’algorithme proposé

Les nœuds procèdent à la capture et chacun envoie individuellement la valeur capturée sous forme de paquet en destination du puits.

Les résultats sont sauvegardés au niveau du puits, dans tableau (tableau 4.2) :

Nœuds	Température (C ⁰) Instant T ₀	Température (C ⁰) Instant T ₁	Humidité (%) Instant T ₂
S ₁	14	16	55
S ₂	16	17	57
S ₃	15	18	58
S ₄	16	18	58
S ₅	15	17	56
S ₆	20	23	47
S ₇	19	22	48
S ₈	17	19	50
S ₉	18	21	49
S ₁₀	19	21	48

TABLE 4.2 – Les résultats de la collecte sans l’algorithme de segmentation de données

4.8.3 Déroulement de l’exemple avec l’algorithme de segmentation de données

Phase de pré-exécution

Au niveau de chaque nœuds, on maintient une vue, et on définit un tableau des classes de données prédéfinie.

Les classes de données de la température	Les bornes	Situations	Décision
C ₁	[5..9]	Pas de feu	Ne rien faire
C ₂	[10..14]		
C ₃	[15..19]		
C ₄	[20..24]		
C ₅	[25..29]		
C ₆	[30..34]		
C ₇	[35..39]		
C ₈	[40..44]	Possibilité qu'il ya un feu	Vérification
C ₉	[45..49]		
C ₁₀	[50..54]		
C ₁₁	>55	Détection d'un feu	Déclenché une alerte

TABLE 4.3 – Les classes de données

Phase d'exécution

– La phase d'initialisation

A l'instant T_0 les capteurs extrêmes captent des valeurs et mettent à jours leurs vues au niveau de chaque capteur ensuite ces valeurs seront envoyés aux nœuds parent (les agrégateurs) qui captent à leurs tour des valeurs et appliquent la fonction d'agrégation (supposons que la fonction est la moyenne AVG).

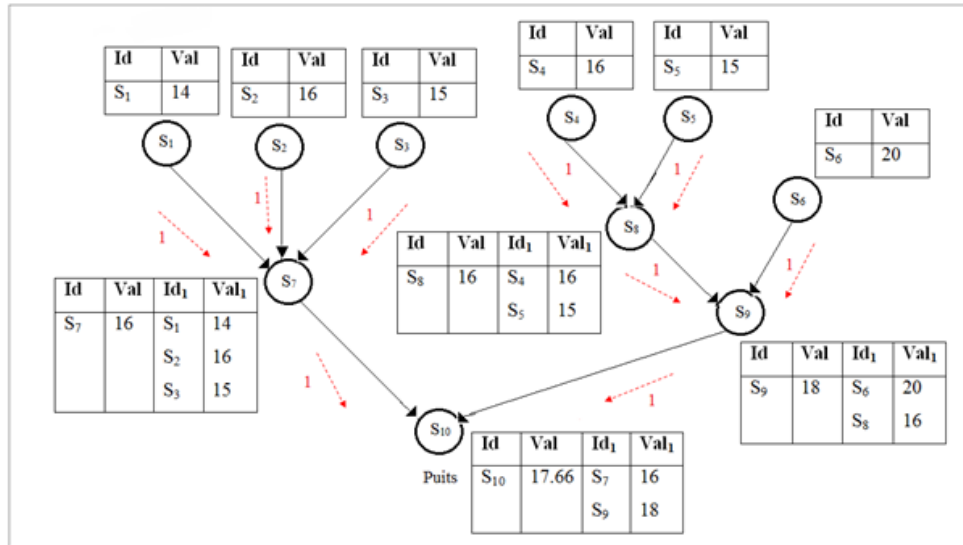


FIGURE 4.3 – Circulation des paquets de données pendant la phase de collecte à l’instant T0 (avec l’algorithme proposé).

On remarque que S_7 transmet uniquement un seul paquet (qui contient les valeurs captée par S_7 , S_1 , S_2 , S_3) vers S_{10} au lieu de trois (04) (un paquet contient 4 données). Egalement S_8 transmet un seul paquet au lieu de 03 (qui contient les valeurs captées par S_9 , S_4 , S_5). S_9 transmet un seul paquet au lieu de 03 (données de S_9 , S_8 , S_6). Soit un total de 9 paquets au lieu de 15 paquets.

– **La phase de collecte**

A l’instant T_1 , une nouvelle requête de collecte est initiée par le puits. Tout se déroule de la même manière, sauf que cette fois, les données captées sont vérifiées au niveau de chaque nœud si elles appartiennent aux mêmes classes de données des valeurs stocké dans les vues. Le résultat est que seulement le capteur S_1 qui n’appartient pas, alors la valeur captée de ce nœud sera transmise à son nœud parent. Au niveau des autres nœuds, ils n’envoient pas leurs valeurs captées. La figure 4.3 montre la circulation des paquets sur le RCSF.

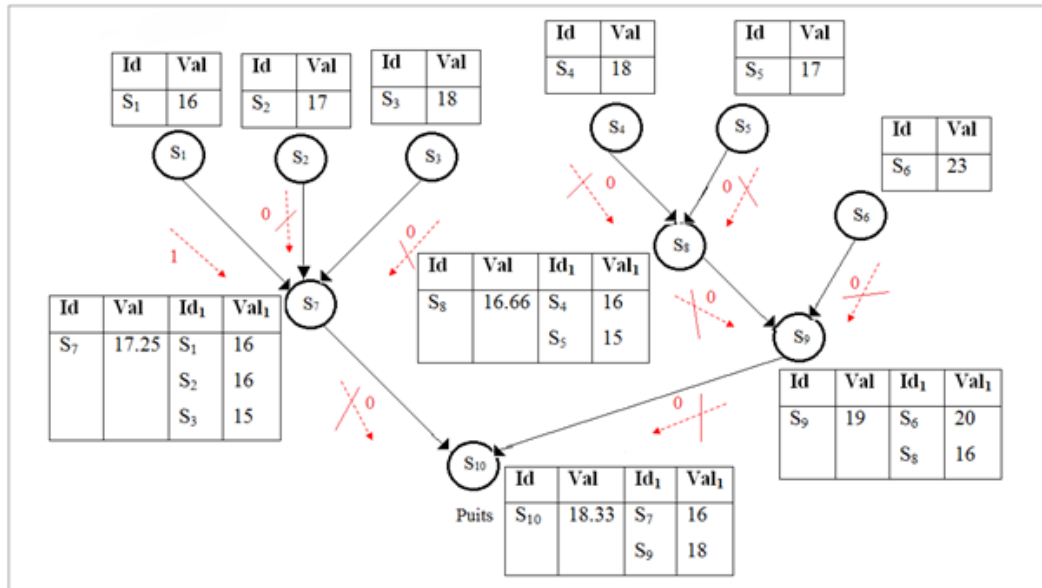


FIGURE 4.4 – Circulation des paquets de données pendant la phase de collecte à l’instant T1 (avec l’algorithme proposé)

Remarquons que cette fois ci, uniquement 01 seul paquet est transmis sur le réseau : S_1 ver S_7 contenant la valeur captée par S_1 .

Le déroulement de la collecte avec l’algorithme fait ressortir l’envoi de 10 paquets (9 à l’instant T_0 et 01 à l’instant T_1) au lieu de 30 paquets en déroulant la collecte sans l’algorithme.

4.8.4 Déroulement de l’algorithme de l’approche améliorée

Nous supposons que le domaine d’application utilisée et le même " domaine environnemental ", et le phénomène physique observé est la détection d’incendie donc le 1^{er} type de données observé est la température, et le 2^{ème} type de donnée à observé est l’humidité.

On définit les classes de données du 2^{ème} type :

Les classes de données de l'humidité	Les bornes	Situations	Décision
C ₁	[10..14]	Humidité trop basse	Assèchement (déclenché un alerte : possibilité qu'il ya un feu)
C ₂	[[15..19]		
C ₃	[20..24]		
C ₄	[[25..29]		
C ₅	[30..34]		
C ₆	[35..39]		
C ₇	[40..44]		
C ₈	[45..49]	Humidité idéal	Ne faire rien
C ₉	[50..54]		
C ₁₀	[55..59]		
C ₁₁	[60..64]		
C ₁₂	[65..69]		
C ₁₃	>70	Humidité trop élevé	Déclenché des pathologies telles que : développement d'acariens, moisissures, condensation, etc.

TABLE 4.4 – Les classes de donnée de type de données humidité

Au niveau de chaque capteur on ajoute une variable H pour stocké la nouvelle valeur collecté d'humidité, et au niveau des nœuds agrégateur on ajoute une variable H1 pour stocké les données d'humidité transmit par les nœuds fils.

A l'instant T₂ une nouvelle requête de collecte est initiée par le puits, ce dernier demande les données du 2^{ème} type qui est l'humidité de l'air. Tout se déroule de la même manière, sauf que cette fois, on envoie avec les données collecté les valeurs des températures qui ne sont pas envoyé à l'instant T₁ pour être encore plus précis. La figure 4.5 montre la circulation des paquets sur le RCSF.

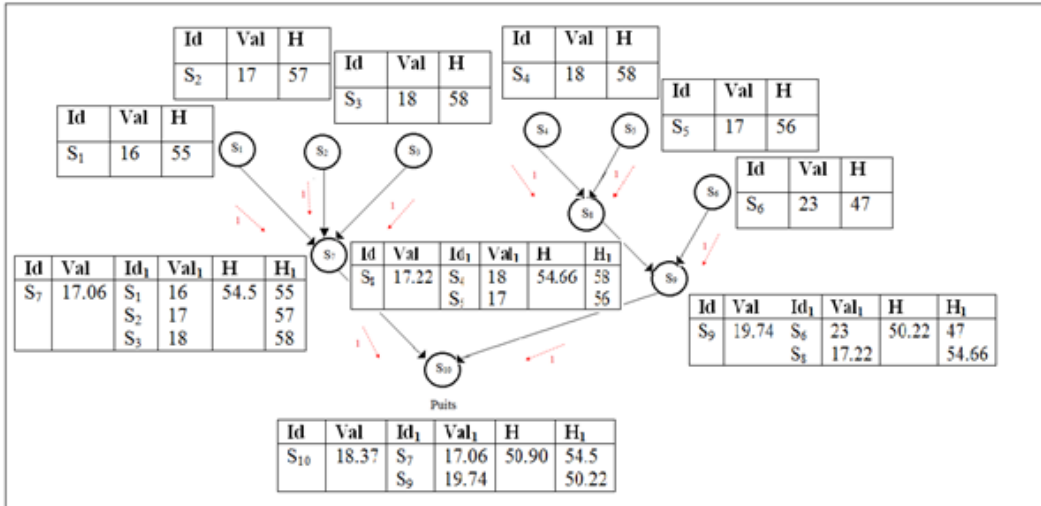


FIGURE 4.5 – Circulation des paquets de données pendant la phase de collecte à l’instant T_2 (avec l’approche améliorée)

4.9 Conclusion

Dans ce chapitre nous avons décrit un algorithme pour agréger les données dans le réseau de capteurs sans fils. L’algorithme se base sur une vue matérialisée pour stocker des données, et l’envoi uniquement de données utiles vers le puits à chaque période. Après le déroulement de l’algorithme nous avons constaté une réduction importante de messages transmis dans le réseau. Ceci nous permet d’économiser l’énergie consommée à cet effet.

CHAPITRE 5

SIMULATION ET EVALUATION DE PERFORMANCES

5.1 Introduction

L'utilisation d'un réseau réel pour l'évaluation est difficile et coûteuse. Aussi, le réseau réel n'offre pas la souplesse de varier ses différents paramètres et pose en plus le problème d'extraction de résultats, c'est pour cela que la majorité des travaux d'évaluation des performances utilisent le principe de la simulation vu les avantages qu'elle offre.

La simulation consiste à gérer le temps ainsi que les actions qui sont liées aux différents instants d'un système réel et à faire fonctionner abstraitement le modèle qui représente ce système.

Afin d'évaluer les performances de notre proposition, nous avons choisi d'utiliser notre propre outil de simulation.

Dans ce chapitre, nous fournissons les résultats de simulation obtenus par notre approche de segmentation de données. Pour évaluer ses performances, nous avons choisi de les comparer par simulation avec celles des résultats du protocole d'agrégation TAG.

5.2 Métriques de performances

Nous commençons tout d'abord, par préciser les métriques que nous avons jugées intéressantes à étudier :

- **L'énergie consommée** : mesure la quantité d'énergie restante par chaque capteur durant la phase de collecte.

- **Le nombre de messages** : mesure le nombre de messages envoyé par chaque capteur durant la phase de collecte.
- **Durée de vie** : mesure la durée de vie de chaque capteurs durant la phase de collecte.

5.3 Modèle de simulation

L'ordinateur utilisé pour l'expérimentation était une unité centrale de traitement Intel (R) Core(TM) i3-2328M CPU avec 2.20 Gigahertz fonctionnant avec 2Go de la RAM et du système d'exploitation Windows 7 professionnel. Le programme a été développé en Java en utilisant JDK/JRE 2.0.7. Nous réalisons la simulation sur un réseau de capteur. Nous exécutons notre algorithme de vérification de classe de différentes valeurs détectées ensuite nous exécutons l'algorithme TAG. L'objectif est de savoir quel est l'impact de notre proposition (vérification de classe de chaque valeur détectée au niveau de chaque capteur) sur le nombre de messages qui circulent sur le réseau et l'énergie.

5.3.1 Choix du langage de programmation

Afin de valider expérimentalement l'algorithme proposé nous avons choisi de concevoir notre propre simulateur en utilisant Eclipse IDE. Eclipse IDE est un environnement de développement intégré libre extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java, et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

5.3.2 Description du système

Notre système représente une zone de capture comportant trois niveaux ; le premier niveau représente les nœuds extrêmes qui ont pour but de surveiller et de détecter les événements qui s'y produisent lors de la réception de la requête, le deuxième niveau qui représente les nœuds agrégateurs qui sont chargé de la détection et l'agrégation et une station de base au troisième niveau, de ce fait, les entités de notre système, ainsi recensées, sont :

- **La station de base** : elle est responsable de l'envoi des requêtes et de la collecte des données du réseau et la réponse aux utilisateurs.
- **Les Nœuds capteurs** : ils servent à détecter les événements.

On a utilisé deux types de capteurs :

Les capteurs extrêmes : ils servent à détecter les événements lors de la réception des requêtes et les envoyer au nœud père.

Les capteurs agrégateurs : ils servent aussi à détecter les événements

Le modèle réseau utilisé consiste en un ensemble de capteurs sans fil, immobiles, aléatoirement dispersés sur le terrain simulé dont l'énergie initiale de chacun d'entre eux est fixée à 50 joules.

5.3.3 Modèle d'énergie

Le modèle radio (voir figure 5.1) proposé par Heinzelman et al. [33] a été utilisé pour calculer l'énergie consommée en émission et en réception des messages. Selon ce modèle l'énergie consommée pendant la transmission (E_e) d'un message de p_k bits est donnée par :

$$E_e = E_{elec} + E_{amp} \times p_k \times d^2$$

Alors que l'énergie consommée pendant la réception (E_r) d'un message de p_k bits est donnée par :

$$E_r = E_{elec} \times p_k.$$

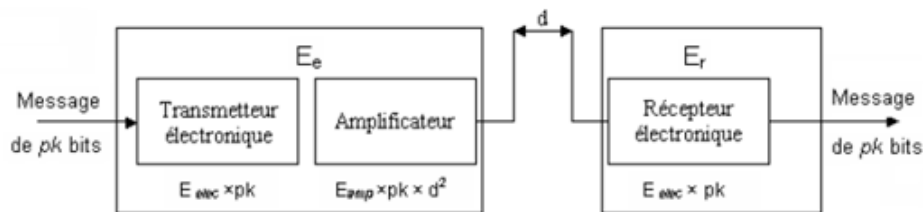


FIG. 4.6 – Modèle d'énergie

FIGURE 5.1 – Modèle d'énergie.

Où :

- E_e est l'énergie consommée en émission.
- E_r est l'énergie consommée en réception.
- E_{elec} est l'énergie électronique (énergie consommée par le transmetteur électronique) dont la valeur est fixée à 100 pJ/bit.
- E_{amp} est l'énergie nécessaire pour l'amplification (l'énergie consommée par l'amplificateur de l'émetteur) dont la valeur est fixée à 50 nJ/bit.
- p_k est la taille en bit d'un paquet de données.

- d est la distance en mètres entre l'émetteur et le récepteur d'un paquet de données.

5.3.4 Etapes de réalisation du simulateur

Les étapes de réalisation de notre simulateur sont les suivantes :

- **Initialisation des variables**

cette étape correspond à la déclaration des variables globales (nombre de capteurs), leur initialisation, la création des capteurs (capacité énergétique) et leur déploiement sur le terrain simulé.

- **Réalisation du simulateur**

En prenant 6 capteurs déployés dans l'environnement ; le prototype est comme suite :

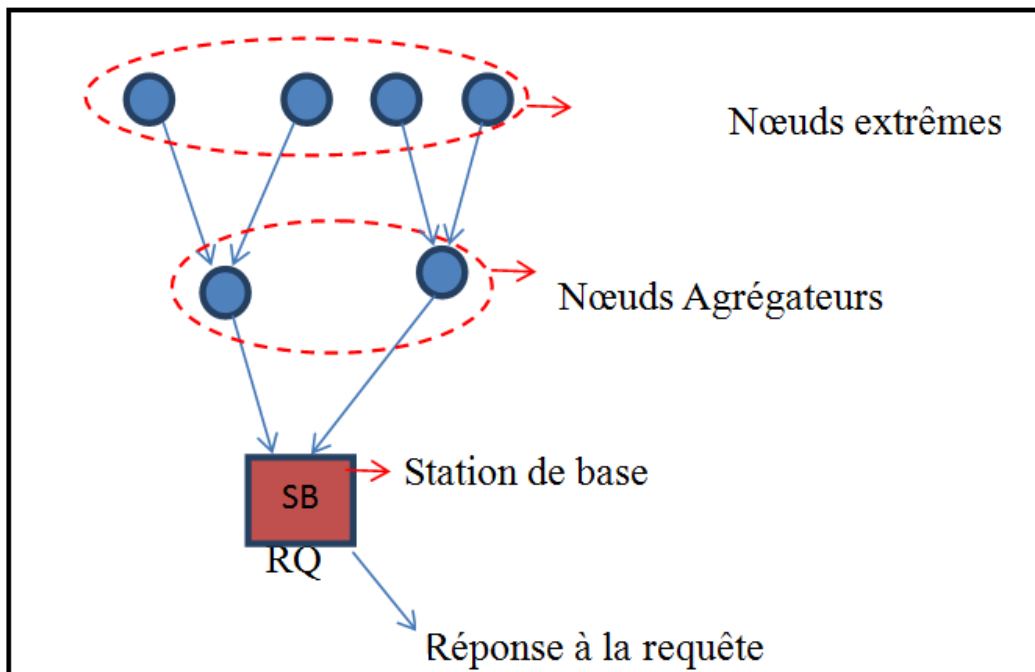


FIGURE 5.2 – Exemple illustratif du prototype

EN utilisant RMI, quand la station de base envoie la requête (à chaque 10 minute par exemple), au nœud puits, qui diffuse la requête à tous les nœuds et récolte toutes leurs réponses, ce cas n'est que celui du client serveur dont chaque deux nœud, le nœud père et le nœud fils s'exécutent comme client et serveur respectivement, de sorte que chaque nœud père envoie la requête à son nœud fils et le nœud fils ne peut pas lui renvoyer la récolte jusqu'à ce qu'il fasse de même c.-à-d. il envoie la requête en tant que client à tous ses nœuds fils (qui sont des serveurs du même niveau) en fin si ces derniers sont des nœuds extrêmes dans ce cas le capteur agrégateur agrège leurs récoltes avec la sienne et la renvoie à son nœud père qui est le client ainsi de suite jusqu'au nœud puits (station de

base) qui répond à l'utilisateur.

5.4 Evaluation de performances

Dans ce qui suit, nous allons présenter et analyser les résultats de simulations obtenus suivant les métriques discutées précédemment. Pour chaque test, 14 exécutions indépendantes sont réalisées : les résultats présentés moyennent les résultats de toutes les exécutions.

Pour les simulations réalisées, le paramètre variable utilisé est la requête envoyé à partir de l'utilisateur à tous les capteurs : augmenter les requêtes vers un capteur équivalent à augmenter les transmissions des données dans le réseau.

5.4.1 Nombre de messages

Nous nous intéressons au nombre de messages générés par les deux algorithmes. Nous recherchons la minimisation du nombre de messages.

On constate que le nombre de messages envoyés, comme le montre la figure 5.3, augmente avec l'augmentation du nombre des requêtes adressées aux capteurs. Ceci est en effet attendu en raison de l'envoi des données à chaque réception d'une requête.

L'approche de TAG permet de générer un nombre de messages sensiblement supérieur à celui obtenu avec l'algorithme de segmentation de données, ceci est expliquée par l'envoi des messages que lorsque la vérification est faite..

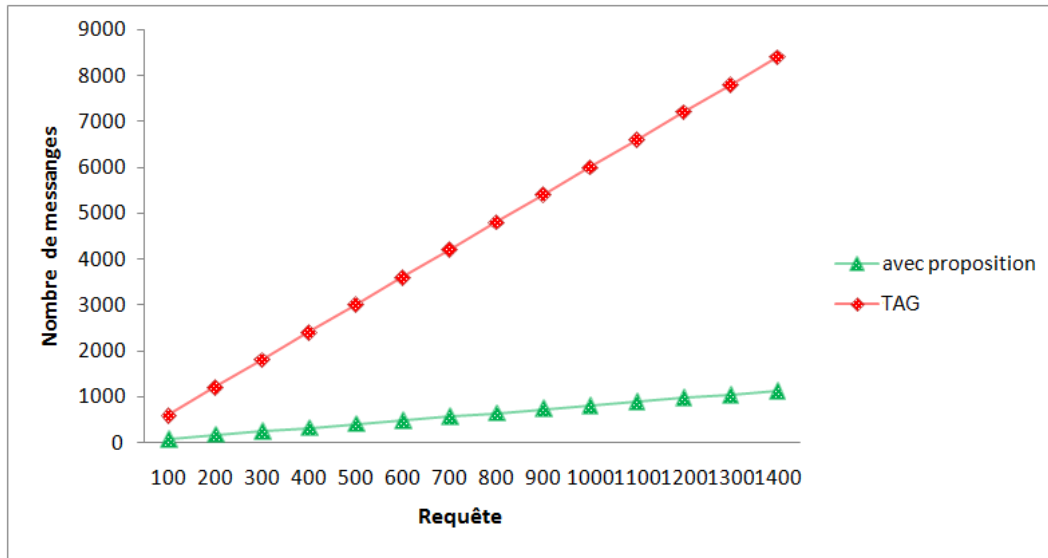


FIGURE 5.3 – Variation du nombre de messages circulant dans le réseau en fonction des requêtes.

5.4.2 Énergie consommée

Vu les limitations en énergie des capteurs, il est primordial de faire en sorte de réduire l'énergie consommée à tous les niveaux et ce afin de permettre une plus grande longévité au réseau.

La figure 5.4 montre l'énergie restante dans le réseau (la somme de toutes les énergies restantes dans chaque capteur) à chaque réception d'une requête avec les deux algorithmes (algorithme de segmentation de données et l'algorithme TAG). On constate, qu'avec le prototype avec segmentation de données de classe et TAG, les capteurs consomment en moyenne la même quantité d'énergie et la même distance séparant les capteurs deux à deux et que l'énergie consommée augmente avec le nombre de requêtes diffusé dans le réseau. La décroissance de la quantité de l'énergie restante avec l'augmentation du nombre de requêtes est attendu car en augmentant le nombre de requêtes à un capteur on augmente le nombre de transmissions dans le réseau ce qui conduit à l'augmentation du nombre de messages échangés entre les capteurs (dont dépend l'énergie consommée en émission et en réception).

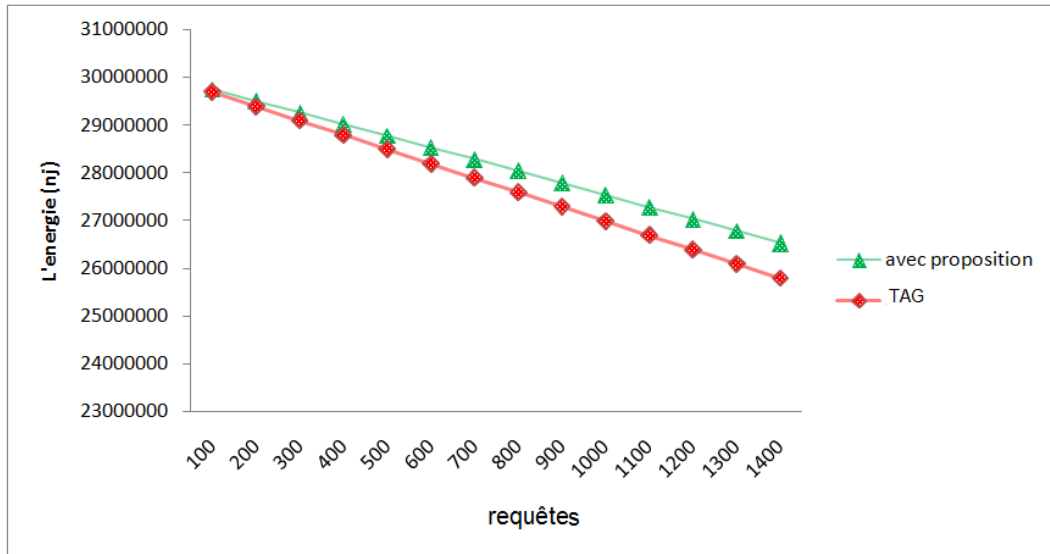


FIGURE 5.4 – Variation d'énergie résiduelle en fonction des requêtes.

5.4.3 Durée de vie

Sachant que l'énergie restante reflète essentiellement la durée de vie du capteur, il est important d'augmenter la durée de vie des capteurs en fonction des requêtes.

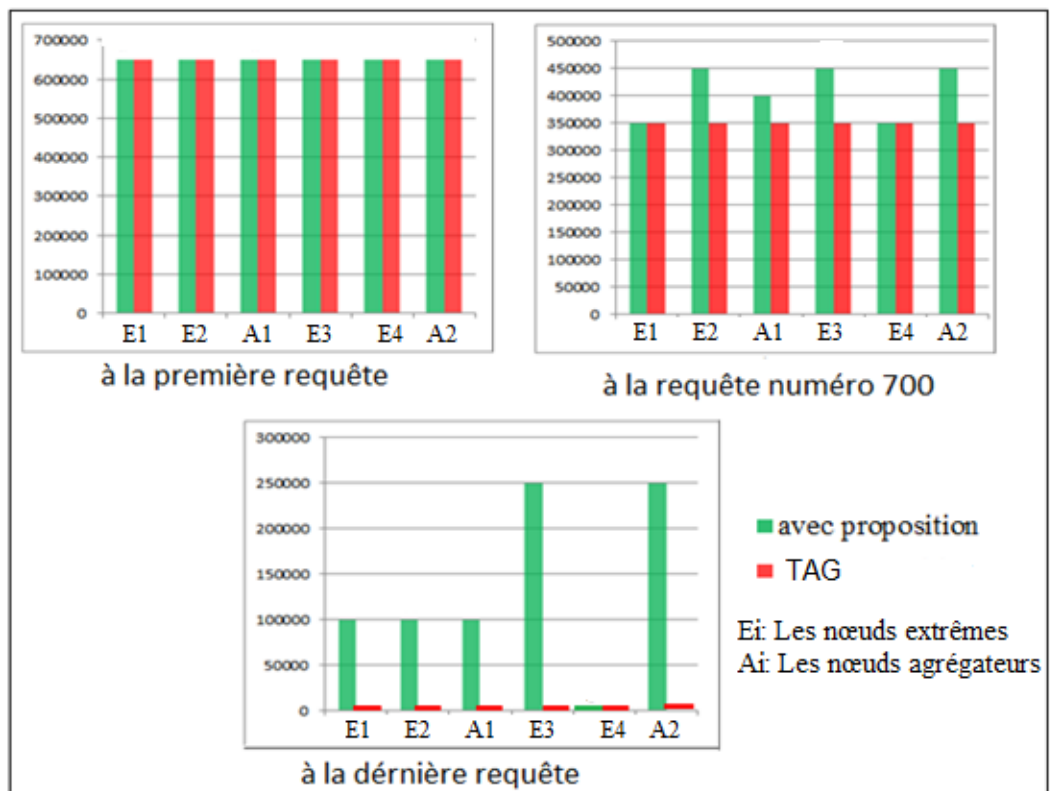


FIGURE 5.5 – Distribution de la durée de vie de chaque capteur en fonction des requêtes.

En optant pour un tel principe, nous avons pu prolonger la durée de vie d'un capteur par requête et ce en faisant en sorte qu'un capteur ayant plus d'énergie restante lors d'une requête donnée privilégie sa durée de vie.

Il est à noter que l'abaissement considérable de l'énergie que peut consacrer chaque capteur au sein du réseau avec TAG. La réduction de la quantité d'énergie avec l'algorithme avec segmentation de données est, quant à elle, est plus faible. Ceci se traduit, comme le montre la figure ci-dessus, par la première requête, où tous les capteurs dispose d'une même quantité d'énergie. A la requête numéro 700 l'énergie restante est réduite d'une même quantité pour tous les capteurs avec TAG et elle est d'une faible réduction avec l'algorithme avec segmentation de données et à la dernière requête, l'énergie est épuisée au niveau des capteurs sans vérification de classe c'est-à-dire que la durée de vie est expirée, par contre dans le cas de vérification de classe il reste encore des capteur à une durée de vie considérable.

5.5 Conclusion

Nous avons pu à travers ce chapitre mettre l'accent sur les principaux points constituant l'approche basée sur la segmentation et la vérification de classe. Le but de cette dernière étant d'assurer une plus grande longévité dans un réseau de capteurs sans fil tout en optimisant d'autres ressources (mémoires et processeurs).

Afin de valider l'approche proposée, des simulations à événements discrets sont réalisées dans le but de montrer l'apport de l'approche de segmentation de données par rapport aux approches déjà existantes. D'après les tests réalisés, notre algorithme est efficace pour réduire le nombre de messages, augmenter la durée de vie, et de mieux économiser les ressources des capteurs en énergie.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Les réseaux de capteurs constituent un axe de recherche très fertile et peuvent être utilisés dans plusieurs domaines différents. Cependant, il reste encore de nombreux problèmes à résoudre dans ce domaine afin de pouvoir optimiser leur utilisation dans le monde réel. L'un des problèmes qu'on peut rencontrer dans ce genre de réseau est la consommation d'énergie qui est une ressource critique dans les réseaux de capteurs causée par l'envoi des données redondantes du même capteur.

Dans ce travail, nous avons proposé et simulé une approche pour gérer les données en utilisant la technique d'agrégation des données. Cette approche permet l'envoi des données utiles vers le nœud puits à chaque période. Cette approche offre un grand avantage dans le contexte de la consommation d'énergie.

La première partie de ce mémoire a été consacrée à l'étude de l'état actuel des choses, de ce qui fait la force des réseaux de capteurs sans fils, de leurs inconvénients. Nous avons aussi cherché à voir dans quel monde évoluent les réseaux de capteurs sans fils, et quelles sont les solutions qui ont déjà été proposées pour améliorer leurs performances.

Nous avons ensuite proposé une nouvelle approche qui permet l'envoi des données utiles dans le réseau de capteurs sans fils grâce à la segmentation des données en classes de permettant, entre autre, un envoi direct des données dans le cas d'appartenance aux classes de données stockées dans chaque capteur, tout en offrant la possibilité de préserver les données captées dans le réseau grâce à une vue matérialisée stockée dans chaque capteur.

L'approche de segmentation de données a été validée expérimentalement. Au vu des performances obtenues par simulation, en montrant la réduction considérable du nombre

de messages envoyé, et en montrant une réduction énergétique.

Comme perspective de notre travail, nous allons continuer à améliorer ce travail, en implémentant un protocole de routage et surtout en prenant en compte d'autres métriques comme la mémoire, et nous allons procéder à la simulation de l'approche améliorée, pour obtenir de meilleurs résultats possible. Nous les envisageons dans nos futurs travaux de recherche qu'ils soient d'ordre individuels, professionnels ou académiques.

BIBLIOGRAPHIE

- [1] C. CASTELLUCCIA et A. FRANCILLON, "Protéger les réseaux de capteurs sans fil", Actes du symposium SSTIC08, pp 1-11, France, 2008.
- [2] Y.YOUNES, " Minimisation d'énergie dans un réseau de capteurs ", Mémoire de magistère : Système informatique, Université Mouloud Mammeri, Algérie, 2012.
- [3] M. AISSANI, "Optimisation du routage dans les réseaux de capteurs pour les applications temps-réel ", thèse de Doctorat en Informatique : Informatique Mobile, l'Université Paris-Est, France, 2011.
- [4] N.TABBENE et M.AFIF, "Routage avec QoS temps réel dans les réseaux de capteurs", mémoire d'Ingénieur en Télécommunications, Option : Ingénierie des réseaux, Ecole Supérieure des communications, Tunisie, 2007.
- [5] M.S. EUGENE WILLIAM, "Conception d'une architecture hiérarchique de réseau de capteurs pour le stockage et la compression de données", Thèse de Doctorat en Informatique, Université de Franche-Comté, France, 2010.
- [6] K.BEYDOUN, "Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs", Thèse de Doctorat en Informatique, Université de Franche-Comté, France, 2009.
- [7] T.BELLACHE et S. BENCHARIF, "Stratégie MIMO coopérative pour l'économie d'énergie dans les réseaux de capteurs sans fil", mémoire de Master en Informatique, Université Abderrahmane Mira de Bejaïa, Algérie, 2012.
- [8] Y.CHALLAL, "Réseaux de Capteurs Sans Fil. Cours, Systèmes Intelligents pour le Transport", Université de Technologie de Compiègne, France,2008.
- [9] K.MEDJHOUM et J.BOUKHOUBZA, "Les Système Embarqué TinyOS", France.

- [10] L.MOUNIER et M.DAMOU "Simulation d'un réseau de capteurs avec TinyOS", Actes de la journée Magistère Informatique Université Joseph Fourier UFR IMA, France, 2007.
- [11] D.REY, "Collecte de données d'un réseau de capteurs sans fils en utilisant une surcouche réseaux pair à pair", en Maîtrise des sciences appliquées : génie informatique, Ecole polytechnique de Montréal, Canada, 2010.
- [12] R.KACIMI, " Techniques de conservation d'énergie pour les réseaux de capteurs sans fil", Thèse de Doctorat : Réseaux et Télécommunications, Université de Toulouse, France, 2009.
- [13] F.AKYILDIZ, W.SU, Y.SANKARASUBRAMANIAM, et E.CAYIRCI, "A Survey of Sensor Network Applications". University of Southern California, USA, 2002.
- [14] A.Makhoul, "Réseaux de capteurs : localisation, couverture et fusion de données", Thèse de doctorat en informatique, Université de Franche-Comté, France, 2008.
- [15] S. MOAD. " Optimisation de la consommation d'énergie dans les réseaux de capteurs sans fil". Mémoire de Master recherche informatique. Université : FSIC-Rennes 1, Laboratoire de recherche : DYONISOS-IRISA, 2008.
- [16] M. LEHSAINI, " Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique ". Thèse de doctorat en Informatique, Université de Franche-comté et Université A.B, France, Tlemcen, 2009.
- [17] M.DEFOSSEUX, " Conception et caractérisation de micro générateurs piézoélectriques pour microsystèmes autonomes ". Thèse de doctorat : MICRONANO ELECTRONIQUE, Université de GRENOBLE, France, 2011.
- [18] E.M.PETRIU, N.D.GEORGANAS, D.C.PETRIU, D.MAKRAKIS et V.Z.GROZA. "Sensor-based information appliances", Instrumentation Measurement Magazine, IEEE, vol.3, No.4, pp.31 - 35, University of Ottawa, Canada, 2000.
- [19] M. TALBI Saïd, " Développement d'un protocole de routage pour les réseaux de capteurs ", Mémoire de Magistère : Réseaux et Systèmes Distribués, Université de Abderrahmane Mira, Béjaïa, 2008.
- [20] W.ZNAIDI, M. MINIER et C.LAURADOUX, " Une proposition d'agrégation de MAC pour les réseaux de capteurs utilisant des fonctions de hachage universelles ", Laboratoire CITI, INSA de Lyon - 6, avenue des arts - 69621 Villeurbanne Cedex - France, UCL / INGI / GSI - Place Saint Barbe, 2 - Louvain-la-Neuve - Belgique, 2009.
- [21] A.GHADDAR, T.RAZAFINDRALAMBO, I.SIMPLOT-RYL, D.SIMPLOT-RYL, et S.TAWBI, " Algorithmes pour l'estimation des données dans les réseaux de capteurs ", inria-00384833, 2009.

-
- [22] B.KRISHNAMACHARI, D.ESTRI, et S.WICKER, "Impact of Data Aggregation in Wireless Sensor Networks", ICDCSW '02 Proceedings of the 22nd International Conference on Distributed Computing Systems pp. 575-578, 2002.
- [23] X.YOUNG, A.ANDRES, G. ANDRES et B.MICKAEL , " Agrégation de données dans les réseaux de capteurs ", 2010.
- [24] G.DE SOUSA, " Etude en vue de la réalisation de logiciels bas niveau dédiés aux réseaux de capteurs sans fil : microsystème de fichiers ", Thèse de doctorat : informatique, Université Blaise Pascal - Clermont II, France, 2008.
- [25] R.RAJAGOPALAN et P.K.VARSHNEY, "Data aggregation techniques in sensor networks ", Syracuse University, USA, Electrical Engineering and Computer Science. Paper 22, 2006.
- [26] L.KHELLADI, N.BADACHE, " Les réseaux de capteurs : état de l'art ", Algérie, 2004.
- [27] M.LEE, J.CHOI et S.SONG, "A Materialized View Approach to Support Aggregation Operations over Long Periods in Sensor Networks", World Academy of Science, Engineering and Technology 20 2008, 2008.
- [28] A. MANJESHWAR et D.P.AGRAWAL, " TEEN : A protocol for Enhanced Efficiency in Wireless Sensor Networks ", Proceeding of the 15th Parallel and Distributed Processing Symp. San Francisco : IEEE Computer Society, pp. 2009-2015. 2001.
- [29] A.MANJESHWAR and D.P.AGRWAL. APTEEN : "A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks", Proceeding of the 2nd Int'l Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing. Florida : IEEE Computer Society, pp.195-202, 2002.
- [30] V.JOLLY et S.LATIFI, " Comprehensive Study of Routing Management in Wireless Sensor Networks- Part-2", 2012.
- [31] C.DINI, "Mécanismes de traitement des données dans les réseaux de capteurs sans fils dans le cas d'accès intermittent à la station de base " Thèse de doctorat en informatique, Université De HAUTE ALSACE, France, 2010
- [32] C.IAANAGONWIWAT, R.GOVINDAN et D.ESTRIN, "Directed diffusion : a scalable and robust communication paradigm for sensor networks", Proceeding MobiCom '00 Proceedings of the 6th annual international conference on Mobile computing and networking, pp. 56-67 , 2000.
- [33] L.PETIT, C.L.RONCANCIO, C.LABBE" Astral : An algebraic approach for sensor data stream querying", Research Report, (RR-LIG-019), LIG, Grenoble, France, 2011.

- [34] E.FASOLO, M.ROSSI, J. WIDMER et M.ZORZI, " In-network Aggregation Techniques for Wireless Sensor Networks : A Survey", DEI, University of Padova, via Gradenigo 6/B -35131, Padova, Italy, DoCoMo Euro-Labs, Landsberger Strasse 312 - 80687 Munich, Germany, 2008.
- [35] J.GEHRKE et S.MADDEN, " Query Processing in Sensor Networks ", University de Cornell, the Massachusetts Institute of Technology, Ithaca, Berkeley, Published by the IEEE CS and IEEE ComSoc, 1536-1268/04/20.00 2004.
- [36] S.R.MADDEN, M.J.FRANKLIN, J.M.HELLERSTEIN et W.HONG, "The Design of an Acquisitional Query Processor for Sensor Networks", In ACMSIGMOD, pp.491-502, 2003.
- [37] Y.YAO et J.E.GEHRKE, "The cougar approach to in-network query processing in sensor networks", In SIGMOD Record, Vol.32, No.3, pp.9-18, 2002.
- [38] A.GUPTA et I.S.MUMICK, "Maintenance of Materialized views : Problems, Techniques and Applications", IEEE Data Engineering Bulletin, Special Issue on Materialized Views and Data Warehousing , Vol.18, pp.3-18, 1995.
- [39] S.MADDEN, M.J.FRANKLIN, J.M.HELLERSTEIN, and W.HONG, "TAG : a Tiny Aggregation Service for Ad-Hoc Sensor Networks",ACM SIGOPSI, Vol. 36, pp. 131-146, 2002.
- [40] D. Z-YAZTI, P. ANDREOU, P. K. CHRYSANTHIS, et G.SAMARAS, "MINT Views : Materialized In-Network Top-kViews in Sensor Networks", Université de Cyprus, USA. Mannheim, pp.182 - 189.
- [41] K-C.K. LEE, W-C.LEE, B. ZHENG, et J.WINTER, "Processing Multiple Aggregation Queries in Geo-Sensor Networks", Univesité de Pennsylvania State, USA et University of Singapore Management, Singapore, DASFAA '06 Proceedings of the 11th international conference on Database Systems for Advanced Applications pp.20-34 , 2006.
- [42] A.ARION, H.JEUNG, K.ABERER, S.MICHEL, "Efficiently Maintaining Distributed Modelased Views on Real-Time Data Streams", Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland.
- a42 N.MEJRI et F.KAMOUN, "Algorithme de Routage Hiérarchique MHEED à Plusieurs Sauts pour Les Grands Réseaux de Capteurs",4th International Conference : Sciences of Electronic, Technologies of Information and Telecommunications, Tunisie, 2007.
- a43 Y.CHALLAL, H.BETTAHAR et A.BOUABDALLAH, "Les Réseaux de capteurs",Université de Technologie de Compiègne, France, 2008.

Résumé

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la gestion des données dans les réseaux de capteurs. Il porte sur l'étude et la mise en œuvre d'un algorithme distribué pour la gestion efficace des données dans les réseaux de capteurs. Dans ce mémoire, nous introduisons une nouvelle approche de gestion de données. L'idée de notre approche est de segmenter les données captées en classes qui seront stockées dans un buffer de chaque capteur, et de maintenir des vues matérialisée stockant les données détectées ou reçues, ainsi notre algorithme permet d'envoyer seulement les nouvelles données qui n'appartient pas à la même classe de la donnée détectée précédemment, une amélioration pour cet approche est proposée dans le cas des capteurs qui détectent des données hétérogènes et qui consiste à profiter d'un envoi d'une donnée pour améliorer la précision des données de natures différentes. L'évaluation des performances de l'approche proposée montre qu'elle assure une réduction considérable d'énergie par rapport aux approches existantes dans la littérature.

Mots-clés : Réseaux de capteurs sans fil, gestion des données, vues matérialisée, Classes de données.

Abstract

The work presented in this paper is lies within the framework of the data management in sensor networks. It relates on the study and implementation of an algorithm distributed for efficient data management in sensor networks. In this paper, we introduce a new approach of data management. The idea of our approach is to segment the captured data into classes that will be stored in a buffer of each sensor, and maintain Materialized views storing detected or received and send only those data that does not belong to the same class of the previously given detected. thus our algorithm enables send only new data that does not belong to the same class of the given previously detected, an improvement for this approach is proposed in the case of sensors that detec heterogeneous data, and that is to take advantage of a send a data to improve the precision of different types of data.

The performance evaluation of the proposed approach shows that it insures a considerable reduction of the in energy compared to existing approaches in the literature.

Keywords : Wireless sensor networks, data management, Materialized view, class of data.