

République Algérienne Démocratique et Populaire

Ministre de L'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE A. MIRA-BEJAIA

FACULTE DES SCIENCES EXACTES

DEPARTEMENT INFORMATIQUE



MEMOIRE DE FIN DE CYCLE

En vue de l'obtention du

Diplôme de Master professionnel en Informatique

Option : Administration et Sécurité des Réseaux

Thème

**La Congestion Dans Les Réseaux De Capteurs Sans Fil
Avec Station Mobile**

Présenté par :

M^{lle} BENARAB Djedjiga

M^{lle} GOUAR Lydia

Encadré par :

M^{me} OUYAHIA Samira

Devant membre de jury :

Présidente : M^{me} SOUADIH Rebiha.

Examinatrice : M^{me} REBOUH Nadjat.

Promotion : 2021-2022

Remerciement

Nos vifs remerciements vont d'emblée à Dieu tout le puissant qui nous a doté d'une grande volonté et d'un savoir adéquat pour mener à bien ce modeste travail

Gratitude particulière et chaleureuse remerciement revient à Notre Promotrice « M^{me} OUYAHIA Samira » qui était toujours là pour nous supporter et nous apporter des précieux et sages conseils et être un support fort grâce à ses orientations riche d'expérience et de professionnalisme avec patience pendant toute la durée de réalisations de ce projet.

Tous les membres de jury d'avoir accepté de juger et d'évaluer notre travail.

Je remercie tous les enseignants du département informatique. Pour tout ce qu'ils nous ont apporté et enseigné.

Un grand merci à tous ceux qui nous ont moralement ou pratiquement soutenus et dont ces lignes ont accidentellement oublié de mentionner.

DEDICACE

A ma Chère Mère Nadia. A mon Père Moussa

*Dont le mérite, les sacrifices et les qualités humaines
m'ont permis de vivre ce jour.*

Mes chères sœurs Adidi, Tounsia, Thelleli.

*Qui n'ont pas cessées de me conseiller, encourager et
soutenir tout au long de mes études. Que dieu les
protéger et leurs offres la chance et le bonheur de la
vie.*

A mes chers Frères : Kamel, Hakim, Athman.

*Pour leurs soutiens, leurs conseils et leurs confiances,
car ils ont toujours cru en moi.*

A mes adorables nièces : Farah, Nihale, Leticia.

*Qui m'ont rendu toujours joyeuses heureuse, que mon
dieu les garde à moi*

A ma copine Hana.

A Mon binôme Lydia.

Djedjiga

DEDICACE

Au meilleur des pères Kaci

A ma très chère maman mira

*Qu'ils trouvent en moi la source de leur fierté A qui
je dois tout*

A mon cher mari Oualid.

*Qui est toujours là pour être l'épaule sur laquelle je
me repose dans mes
Moments de fatigue*

*A ma sœur tinhinane qui occupe une place unique
dans ma vie, et mes*

*Chères frères Amar et Nassim pour leur grande
tendresse et leur amour,*

A qui je souhaite un avenir radieux plein de réussite

A tous ceux qui me sont chers

Lynda, Merieme, Souhila

A mon binôme Djidji

Lydia

LISTE DES FIGURES

Figure 1	Composant d'un capteur sans fil	4
Figure 2	Exemple d'un réseau de capteur sans fil.....	5
Figure 3	Les domaines d'application des réseaux de capteurs sans fil.....	8
Figure 4	La pile protocolaire dans les RCSFs.	11
Figure 5	Un exemple de STTD.....	19
Figure 6	Représentation schématique de la méthodologie	30
Figure 7	Architecture globale du schéma PSOGSA proposé.....	32
Figure 8	L'interface du simulateur.....	37
Figure 9	Importer la topologie	38
Figure 10	Le réseau de 100 nœuds déployés sur le terrain.....	38
Figure 11	Les paramètres concernant le nœud puits	39
Figure 12	Les paramètres concernant les nœuds sources.....	39
Figure 13	Première version de l'arbre pour la position du puits au niveau du nœud 39	39 ..40
Figure 14	Deuxième version de l'arbre pour la position du puits au niveau du nœud 89..	41 41
Figure 15	Troisième version de l'arbre pour la position du puits au niveau du nœud 51	41 41
Figure 16	Résultat de simulation	43

LISTE DES TABLEAUX

Tableau 1 : Les paquets de contrôle et la structure du père et fils.	21
Tableau 2: Résultats de la simulation.....	43

LISTE DES ABREVIATIONS

ACK: ACKnowledgement.

AWF: Adaptive Weight Firefly.

CSMA : Carrier Sens Multiple Access.

CMR: Coordinate Magnetic Routing.

DAIPaS: Dynamic Alternative Path Selection Scheme.

FFA: Firefly algorithm.

GBEER: Grid-Based Energy-Efficient Routing.

HRTC: Hybride Ressource Traffic Control.

MAC: Media Access Control.

OSI: Open System Interconnections.

PSOGSA : : Algorithme de Contrôle de la Congestion basé sur l'Algorithme d'Optimisation Multi-Objectif.

Qos: Quality of Service.

RCSF : Réseau de Capteur Sans Fil.

RCSFs-EM : Réseaux de Capteurs Sans Fil à Eléments Mobiles.

STDD: Sink-oriented Tree based Data Dissemination.

SVM: Support Vector Machines.

TTDD: Two-Tier Data Dissemination in Large-scale Wireless Sensor Networks.

WSN : Wireless Sensor Network

INTRODUCTION GENERALE

Les réseaux de capteurs sans fil (RCSF) sont un large domaine qui est utilisé dans de nombreuses applications ces dernières années. De la surveillance du trafic à la surveillance médicale, environnement et militaire, les RCSF sont utilisés presque dans tous les domaines. Les capteurs contiennent une batterie, un émetteur, un récepteur, une unité de traitement et une unité de captage. Les nœuds capteurs sont généralement déployés dans le cadre de réseaux sans fil pour collecter et transmettre des données à une ou plusieurs stations de base. La station de base mobile dispose d'une puissance de calcul élevée et d'une grande capacité de stockage pour traiter et mémoriser les données reçues.

Dans ce projet, nous allons présenter les contraintes dans les RCSF qui ont donné naissance à des protocoles de routage et de congestion différents que ceux des autres réseaux sans fil. Puisque la contrainte de congestion au niveau des nœuds se pose avec force dans les RCSF et que celle-ci peut être considérée dans le protocole de routage, nous nous sommes intéressées au protocole de routage de diffusion de données (STDD) proposé pour les RCSFs à station de base mobile. Les RCSFs à station de base mobile sont les meilleurs dans l'efficacité à résoudre le problème de surcharge du réseau mais à condition que la transmission des données soit bien équilibrée entre les nœuds relais pour éviter de les congestionner. Ainsi, nous avons étudié ce critère dans STDD et nous avons trouvé qu'il ne prend pas en considération l'équilibrage de charge en transmettant les données à travers les nœuds relais. En effet, nous avons remarqué qu'il y a des nœuds qui sont surchargés et qui peuvent être une source de congestion. La mise en place des techniques capables d'éviter et de contrôler ce type de congestion devient donc une nécessité.

Pour remédier au problème de l'équilibrage de charge et éviter puis corriger les congestions dans le protocole de routage STDD, nous proposons de l'améliorer avec l'utilisation de l'algorithme Firefly qui est une méta heuristique inspirée du comportement social des lucioles dans la nature. Notre objectif dans le cadre de ce mémoire consiste à l'utiliser pour éviter la congestion puis la corriger si elle survient inévitablement dans un réseau de capteur sans fil avec station mobile fonctionnant avec le protocole de routage STDD. L'objectif final est d'améliorer les performances des RCSFs à savoir, réduire la consommation de l'énergie, la perte des données et augmenter la durée de vie du réseau.

Nous avons organisé notre mémoire en trois chapitres.

INTRODUCTION GENERALE

Dans le premier chapitre, nous avons présenté les généralités sur les RCSFs qui vont permettre aux lecteurs de comprendre le contexte de notre projet et la problématique traitée.

Dans le deuxième chapitre, nous avons présenté le problème de la congestion et quelques solutions proposées dans la littérature pour le résoudre. Nous avons aussi présenté le protocole de routage que nous voulons étudier et améliorer.

Enfin, le dernier chapitre a fait l'objet de l'analyse de la congestion dans le protocole STDD et la proposition d'une amélioration pour résoudre le problème de congestion dont il souffre.

CHAPITRE 1

Généralités sur les réseaux de capteurs sans fil

1 Introduction

Le développement technologique au cours des dernières décennies a permis de fabriquer des capteurs de plus en plus petits, de plus en plus performants et avec des autonomies énergétiques grandissantes. Ces derniers sont dotés de capacités embarquées de traitement, de stockage, de détection et de communication sans fil. Ils sont déployés en centaines dans une zone géographique difficile d'accès formant ainsi un Réseau de Capteurs Sans Fil (RCSF) également connu sous le nom de WSN (Wireless Sensor Networks).

Un RCSF (réseau de capteur sans fil) peut alors être défini comme un système autonome et dynamique, composé de nœuds statiques ou mobiles. Ces derniers sont interconnectés par des liens sans fil, sans l'aide d'aucune administration centralisée ni d'infrastructure. Un RCSF a pour tâche principale de prendre les mesures scalaires des phénomènes physiques, tels que la température, la pression et l'humidité ainsi que la localisation et le suivi des objets.

Dans ce chapitre, nous allons présenter les capteurs et leurs composants, nous allons introduire et faire une description synthétique des réseaux de capteurs sans fil (fonctionnement, caractéristiques, et leurs domaines d'applications variés). Ensuite, nous allons discuter également l'aspect mobilité dans les RCSFs. Nous présentons en dernier lieu la pile protocolaire et les plans de gestion.

2 Capteur sans fil

2.1 Définition capteur

Un capteur sans fil est un petit dispositif électronique capable de mesurer une valeur physique environnementale (température, lumière, pression, etc.) et de la communiquer à un centre de contrôle via une station de base [01]. Un nœud capteur sans fil est donc, capable d'accomplir trois tâches complémentaires :

- Le relevé d'une grandeur physique.
- Le traitement de l'information.
- La communication avec d'autres nœuds capteurs.

2.2 Composant d'un capteur sans fil

Un capteur sans fil comporte quatre unités de base représentées par une unité d'acquisition (dispositif de captage), une unité de traitement (un processeur), une unité de

communication (un émetteur/récepteur radio) et une batterie (Voir la **Figure-1**). De plus, un nœud capteur peut être équipé d'autres composants supplémentaires tels qu'un système de localisation géographique GPS (Global Position System). Le rôle de chacune des unités est défini dans les points suivants : [02]

2.2.1 L'unité d'acquisition :

Elle est généralement composée de deux sous-unités, un capteur qui permet d'obtenir des mesures sur les paramètres environnementaux et un convertisseur Analogique/Numérique qui convertit l'information relevée et la transmet à l'unité de traitement [03].

2.2.2 L'unité de traitement :

Cette unité est responsable de tous les traitements que doit effectuer un nœud capteur. Elle comprend deux interfaces : une interface avec l'unité d'acquisition et une autre avec le module de transmission. L'unité de traitement contrôle les procédures permettant au nœud capteur de réaliser les tâches d'acquisition et de stockage de données collectées, à travers un microcontrôleur (un simple processeur) et une mémoire limitée à quelques kilos octets [04].

2.2.3 L'unité de communication :

Elle est composée d'un émetteur/récepteur permettant la communication entre les différents nœuds du réseau via un support de communication radio [05].

2.2.4 Batterie :

Alimente les unités d'acquisition, de traitement et de communication.

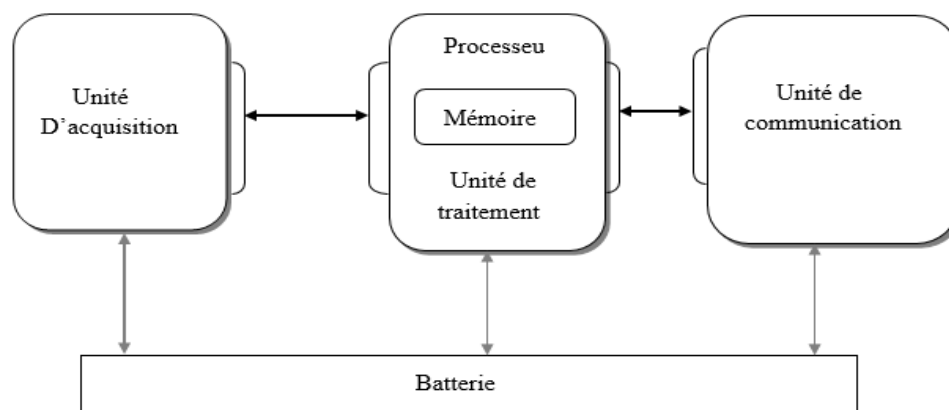


Figure 1 Composant d'un capteur sans fil

3 Définition d'un RCSF ou WSN (Wireless Sensor Network)

Un réseau de capteur sans fil est un type particulier de réseau ad-hoc défini par un ensemble coopérant des nœuds capteurs dispersés dans une zone géographique appelée zone de captage afin de surveiller un phénomène et récolter ses données d'une manière autonome. Le réseau possède en général un nœud particulier, la station de base, connectée avec les autres nœuds. Une architecture plus complète des RCSFs est présentée dans la **Figure 2**. Cette dernière montre qu'un réseau de capteur peut contenir un nœud agrégateur qui se charge de collecter les données de ses voisins et de les acheminer vers le nœud puits. La **Figure 2** illustre aussi que le nœud puits envoie à travers Internet les données collectées par les nœuds capteur vers un utilisateur final ou une base de données [06].

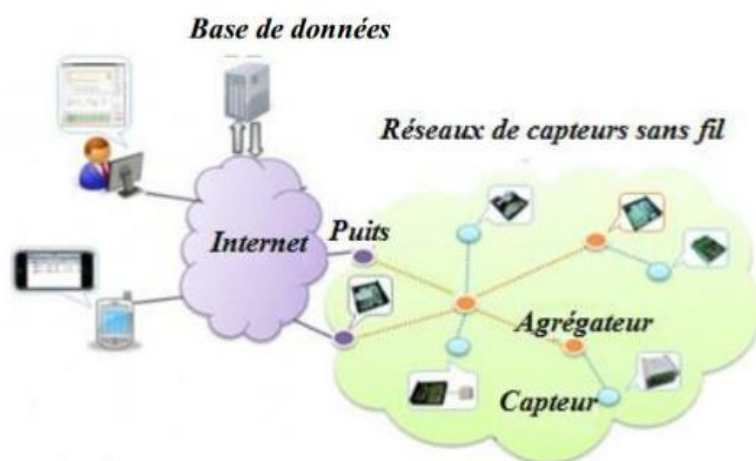


Figure 2 Exemple d'un réseau de capteur sans fil

4 Fonctionnement d'un réseau de capteurs

Les données captées par les nœuds sont acheminées grâce à un routage multi-saut à un nœud considéré comme un "point de collecte" appelé nœud-puits (ou Sink). Ce dernier peut être connecté à l'utilisateur du réseau via internet, un satellite ou un autre système. L'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises pour récolter les données environnementales captées par le biais du nœud-puits [07].

5 Caractéristique des RCSFs

Les caractéristiques essentielles des RCSFs s'établissent comme suit [02] :

5.1 Densité importante des nœuds

Les réseaux de capteurs se composent généralement d'un nombre très important de nœuds pour garantir une couverture totale de la zone surveillée. Ceci engendre un niveau de surveillance élevé et assure une transmission plus fiable des données sur l'état du champ de captage.

5.2 Topologie dynamique

L'instabilité de la topologie des réseaux de capteurs est le résultat des trois facteurs essentiels suivants :

- **La mobilité des nœuds** : les nœuds capteurs peuvent être attachés à des objets mobiles qui se déplacent librement et arbitrairement, introduisant ainsi une topologie instable du réseau.
- **La défaillance des nœuds** : du fait de l'autonomie énergétique limitée des nœuds, la topologie du réseau n'est pas fixe (les nœuds qui épuisent leur énergie, sont considérés comme des nœuds inexistantes).
- **L'ajout de nouveaux nœuds** : de nouveaux nœuds peuvent facilement être rajoutés. Il suffit de placer un nouveau capteur qui soit dans la portée de communication d'au moins un autre nœud capteur du réseau déjà existant.

5.3 Auto-organisation

L'auto organisation s'avère très nécessaire pour ce type de réseau afin de garantir sa maintenance. Vu les différentes conséquences résultant de l'instabilité de la topologie du réseau de capteur, ce dernier devra être capable de s'auto-organiser pour continuer ses applications.

5.4 Scalabilité

Les réseaux de capteurs peuvent contenir des centaines voire des milliers de nœuds capteurs. Un nombre aussi important engendre beaucoup de transmissions inter-nodales et nécessite que le nœud « Sink » soit équipé d'une mémoire importante pour stocker les informations reçues.

6 Les domaines d'application dans les réseaux de capteurs sans fil

La taille de plus en plus réduite des micro-capteurs, le coût de plus en plus faible, la large gamme des types de capteurs disponibles (thermique, optique, vibrations) ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs d'envahir plusieurs domaines d'application, comme présenté dans la **Figure 3**. Ils permettent aussi d'étendre les applications existantes et de faciliter la conception d'autres systèmes tels que le contrôle et l'automatisation des chaînes de montage. Parmi les domaines où ces réseaux peuvent offrir les meilleures contributions, nous citons : [02]

6.1 Applications militaires

Les nœuds capteurs devraient fournir les services suivants :

- Surveillance des champs de bataille.
- Reconnaissance des forces d'opposition.
- Repérage des cibles.
- Évaluation des dommages de la bataille.
- Détection et reconnaissance d'attaque nucléaire, biologique et chimique.

6.2 Applications de santé

- Certaines applications de santé des RCSF sont :
- Fourniture d'interfaces pour les handicapés.
- Repérage et surveillance des médecins et des patients dans les hôpitaux.
- Télésurveillance des données physiologiques humaines.

6.3 Applications environnementales

Ces applications incluent :

- Le repérage des mouvements des oiseaux, des petits animaux et des insectes.
- La surveillance des conditions environnementales qui affectent les récoltes et le bétail.
- L'exploration planétaire.
- Alertes des catastrophes (incendie, séisme. . .).
- La détection d'inondation.
- L'étude de pollution.

6.4 La domotique

Un autre type d'application dans lequel les réseaux de capteurs émergent, est la domotique. Dans cette application, le réseau de capteurs est déployé dans l'habitation. Le principe est que le réseau forme un environnement, dit pervasif où l'objectif étant de fournir toutes les informations nécessaires aux applications d'automatisation pour le confort, la sécurité et la maintenance dans la maison.

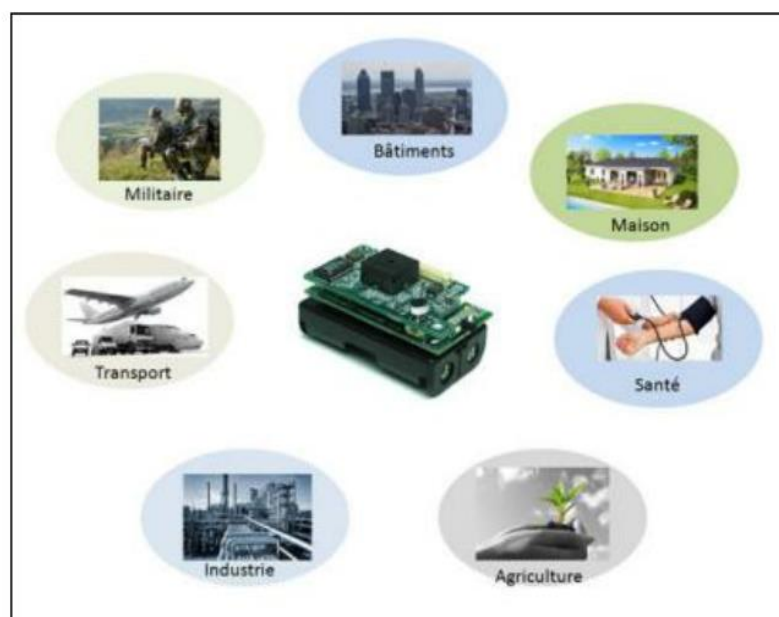


Figure 3 Les domaines d'application des réseaux de capteurs sans fil.

7 Réseaux de capteurs sans fil à éléments mobiles (RCSFs-EM)

Les réseaux de capteurs sans fil peuvent être distingués selon la mobilité des nœuds qui les composent. Les RCSF comprennent des nœuds mobiles qui se déplacent pour interagir avec leur environnement. La conception des RCSFs mobiles inclut le modèle de déploiement initial, le modèle de mobilité, les techniques de localisation et de contrôle. Elle a aussi pour objet de minimiser la consommation d'énergie tout en garantissant le maintien de la connectivité et une bonne collecte de données [08].

Un RCSF-EM est à l'origine un RCSF qui est constitué de nœuds, d'une zone d'intérêt, et d'un observateur qui peuvent être mobile. Si les nœuds se déplacent, cela implique un

changement de la zone d'intérêt qui devient alors « mobile ». Comme la zone change, l'observateur sera obligé de reconstruire le chemin d'accès et de remplacer l'ancien. Dans ce cas, l'observateur s'intéresse à construire le chemin entre lui et les nœuds et à choisir le plus optimal. L'objectif des RCSFs-EM est de collecter plus d'informations sur un environnement en utilisant moins de nœuds capteurs, et de permettre au réseau d'organiser lui-même ses nœuds, de mieux assurer la connectivité et de la couverture au sein du réseau et surtout de prolonger la durée de vie du réseau le plus longtemps possible pendant le processus de collecte de données [09].

En outre, il devient possible de déplacer ses capteurs dynamiquement selon les changements environnementaux, ce qui le rend adaptable à l'évolution de son environnement.

7.1 Différentes formes de mobilité

Nous pouvons distinguer trois formes de mobilité dans les RCSFs

7.1.1 Mobilité des nœuds capteurs :

Peu d'applications sont concernées par la mobilité des capteurs, par exemple des applications militaires (les capteurs sont attachés à des soldats). Dans ce cas, le réseau doit se réorganiser assez fréquemment pour pouvoir fonctionner correctement. Ceci augmente et accentue le problème de la Conservation de l'énergie des nœuds car l'utilisation des techniques de géo localisation augmente et ces dernières consomment beaucoup d'énergie.

7.1.2 Mobilité de l'événement :

Ce type de mobilité existe essentiellement dans les applications de suivi de cibles. Dans ce type d'application, il est important que l'évènement observé soit couvert par un nombre suffisant de nœuds. Par conséquence, les nœuds vont se réveiller autour de l'objet, pour le surveiller avec un taux d'activité élevé, et le nœud capteur se mettra en mode "veille" lorsque la cible s'éloigne. Pendant que la source d'évènement se déplace à travers le réseau, elle est accompagnée d'un secteur d'activité dans le réseau qui la suit.

7.1.3 Mobilité du nœud-puits :

C'est un cas spécial de mobilité où seul le point de collecte est mobile. Etant donné que le trafic converge vers le nœud-puits, les nœuds proches de lui transmettent plus de paquets que

les autres nœuds et donc épuisent plus vite leurs batteries. Si ce nœud-puits est mobile, les nœuds capteurs peuvent attendre son passage pour lui envoyer leurs données directement ou sur un nombre de sauts réduit. Cela va réduire considérablement le trafic et la consommation d'énergie dans les réseaux.

7.2 Architecture d'un RCSF avec nœud-puits mobile

Lorsque le nœud-puits est mobile, il se déplace autour ou à l'intérieur du réseau pour la collecte de données. Un réseau de capteurs sans fil muni d'un nœud-puits mobile est généralement constitué de : [05]

7.2.1 Nœuds capteurs sources :

Déployés dans une zone de captage pour collecter des données ou détecter des évènements.

7.2.2 Nœud-puits mobile :

C'est un dispositif électronique considéré comme un point de collecte de données, il ne possède aucune contrainte énergétique ce qui lui permet de se déplacer autour ou à l'intérieur de la zone de captage à l'aide d'un support mobile (voiture, robot, animal ou humain). Selon l'application, le nœud-puits collecte les données captées par les nœuds soit directement (en parcourant et en collectant les données de chaque capteur dans le réseau) ou bien indirectement à travers des nœuds intermédiaires.

7.2.3 Nœuds relais :

C'est un nœud qui peut avoir (ou non) plus de ressources que les nœuds du réseau et qui a le rôle d'agréger les informations captées de ses nœuds voisins et de les acheminer vers d'autres nœuds relais ou vers le nœud-puits.

7.3 Avantage de l'utilisation d'un nœud-puits mobile

Les réseaux de capteurs sans fil présentent plusieurs contraintes en termes d'énergie, de calcul et de bande passante. Le principal objectif de tels réseaux est d'assurer les communications entre les capteurs sans fil et le nœud-puits et d'éviter les pertes de données qui peuvent influencer sévèrement les performances du réseau.

8 La pile protocolaire

La pile protocolaire est mise en place afin de structurer les protocoles de communication dans les RCSFs. Ce modèle comprend 8 couches (voir la **Figure 4**), cinq d'entre elles ont les mêmes tâches que celles du modèle OSI (Open System Interconnections), et trois autres couches pour assurer la gestion d'énergie (Power Management Plane), la gestion de la mobilité (Mobility Management Plane), et la gestion des tâches (Task Management Plane) [10].

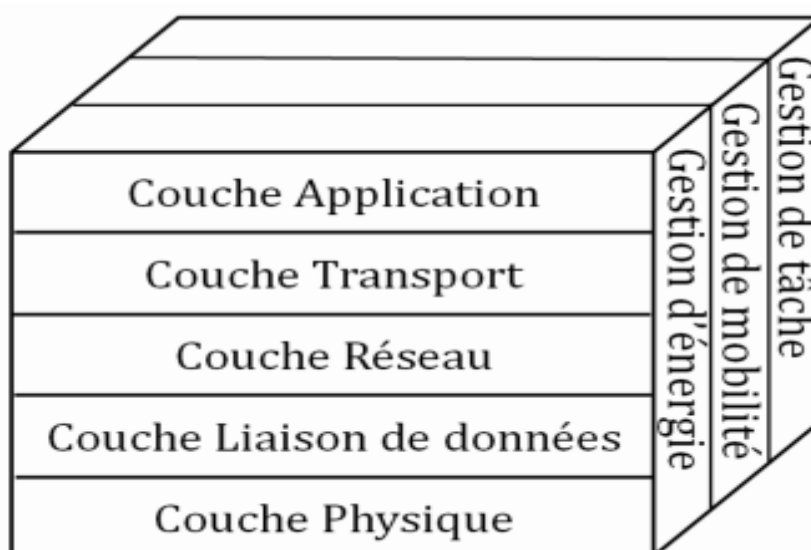


Figure 4 La pile protocolaire dans les RCSFs.

Le rôle de chacune des cinq couches ainsi que les protocoles en vedette sont :

8.1 La couche physique

Cette couche se charge de tout ce qui est spécifications des caractéristiques matérielles, la génération des ondes porteuses, la modulation de données et leur injection sur le support de transmission tout en sélectionnant les bonnes fréquences.

8.2 La couche liaison de données

Spécifie comment les données sont expédiées entre deux nœuds dans une distance d'un saut. Elle est responsable de l'accès au media physique, multiplexage des données et du contrôle d'erreurs. Elle assure la liaison point à point et multipoint dans le réseau [11].

8.3 La couche réseau

La couche réseau a pour but principal de baliser une route optimale en vue d'acheminer efficacement les données captées depuis leur source jusqu'au puits, tout en minimisant la dissipation énergétique des nœuds capteurs inclus dans le chemin. La tâche de routage au sein d'un réseau de capteurs est spécifique du fait que :

- L'écoulement de données récoltées à partir de multiples sources vers une seule destination (la station de base).
- La forte redondance de données et l'exigence de l'agrégation.
- La nécessité d'une gestion soignée des ressources (énergie, mémoire, bande passante).

8.4 La couche transport :

Elle assure la fiabilité et la qualité des données au niveau de la source et au niveau du Sink. Les protocoles de la couche transport pour les RCSF doivent supporter différentes applications, une fiabilité variable et des mécanismes de récupération des paquets perdus et de contrôle de congestion. Le développement d'un protocole pour la couche transport doit être générique et indépendant de l'application. Il doit offrir une fiabilité variable des paquets pour différentes applications.

8.5 La couche application

La couche application présente le niveau le plus proche des utilisateurs. De nombreux profils d'applications peuvent être configurées et utilisées dans la couche application des réseaux de capteurs sans fil.

8.6 Les plans de gestion

Les plans de gestion d'énergie, de mobilité et des tâches permettent au nœud capteur de contrôler respectivement la dissipation d'énergie, le mouvement et la distribution de tâches. En effet, ces plans aident également les nœuds capteurs à coordonner la tâche de captage tout en rationalisant la consommation énergétique. Ils sont donc nécessaires pour que les nœuds capteurs puissent collaborer ensemble pour acheminer les données dans un réseau mobile et partager les ressources entre eux avec une consommation efficace de l'énergie.

8.6.1 Plan de gestion de mobilité :

Offre des mécanismes de détection et enregistrement des mouvements du nœud capteur.

Ainsi, le nœud capteur peut garder trace de ses voisins.

8.6.2 Plan de gestion d'énergie :

La vie du nœud montre une dépendance forte à l'égard de la vie de la batterie. Les fonctions intégrées à ce niveau consistent à gérer l'énergie consommée par les capteurs.

8.6.3 Plan de gestion de tâche :

Assure l'équilibrage et la distribution des tâches sur les différents nœuds du réseau, afin d'assurer un travail coopératif et efficace en matière de consommation d'énergie, ce qui permet de prolonger la durée de vie du réseau.

9 La problématique

Nous avons vu précédemment que l'utilisation d'un nœud-puits mobile dans les RCSFs permet d'économiser l'énergie des nœuds capteurs qui sont proches du puits. En effet, cette solution permet de régler ce qu'on appelle le problème des nœuds chauds ou en anglais « hot spot problem ». Certes, ceci permet d'augmenter la durée de vie du réseau mais, il reste encore des problèmes qui mènent au gaspillage de l'énergie des nœuds capteurs et qui réduisent ainsi la durée de vie du réseau. Parmi ces problèmes, nous citons la congestion des nœuds capteurs. En effet, ce problème cause un gaspillage d'énergie puisqu'il peut y avoir beaucoup de retransmission des paquets supprimés à cause du manque d'espace mémoire dans les files des nœuds congestionnés. Dans le cadre de ce présent projet, nous allons étudier le problème de la congestion dans les RCSFs à station de base mobile et proposer une solution pour augmenter la durée de vie de ces réseaux.

10 Conclusion

Dans ce chapitre, nous avons présenté des généralités sur les réseaux de capteurs sans fil que nous estimons utiles pour comprendre notre travail et la suite de ce mémoire. Nous avons terminé par présenter la problématique traitée dans ce projet pour faire le lien avec le chapitre suivant qui présente un état de l'art sur les solutions proposées pour le problème de congestion dans les RCSFs.

CHAPITRE 2

LES PROTOCOLES DE ROUTAGE ET DE CONGESTION DANS LES RESEAUX DE CAPTEURS SANS FIL

1 Introduction

Les réseaux de capteurs sans fil traditionnels s'appuient sur des nœuds statiques. Ces réseaux peuvent être utilisés dans des applications telles que les applications commerciales, environnementales, les systèmes de surveillance, de repérage, etc. Néanmoins, étant donné que les nœuds dans un réseau de capteurs sans fil sont des dispositifs électroniques, portables et de petite taille, ils peuvent être facilement couplés à des entités mobiles telles que des véhicules, des robots, des animaux ou des personnes. Par conséquent, dans les réseaux de capteurs sans fil, il est essentiel de prendre en charge des mécanismes de mobilité efficaces, sans compromettre le fonctionnement principal de l'application et la durée de vie du réseau. Dans notre projet, nous nous intéressons aux mécanismes de contrôle de congestion

En effet, il existe plusieurs sources de congestion dans les RCSFs, comme le débordement des mémoires tampon, les transmissions concurrentes et la collision des paquets. Parmi les problèmes causés par la congestion, on peut citer : le retard de l'information, perte de paquets qui contiennent parfois les informations critiques, gaspillage de la bande passante et de l'énergie. Il est facile de constater que la congestion dégrade les performances du réseau en l'empêchant de garantir certaines exigences de Qualité de service comme le temps réel, le routage de bout en bout et la maximisation de la durée de vie du réseau. Ce problème motive le besoin de mise en place des mécanismes de contrôle de congestion pour améliorer la performance et prolonger la durée de vie du réseau. Dans les réseaux de capteurs, la plupart des mécanismes de contrôle de congestion sont implémentés au niveau du protocole de routage. Ainsi, l'objectif de ce chapitre est de présenter le routage et le contrôle de congestion dans les RCSFs à travers quelques protocoles proposés dans la littérature.

2 Routage dans les réseaux de capteurs à station de base mobile

Avant de présenter l'état de l'art sur les contrôles de congestion, nous allons définir le routage et présenter ses contraintes dans les RCSFs, puis terminer par présenter un protocole de routage proposé dans pour les RCSFs à puits mobile.

2.1 Définition du routage

Le routage est le mécanisme par lequel des chemins sont sélectionnés dans un réseau pour acheminer les données d'un expéditeur jusqu'à un ou plusieurs destinataires. Sa performance

est importante dans les réseaux décentralisés, c'est-à-dire où l'information n'est pas distribuée par une seule source, mais échangée entre des agents indépendants [12].

2.2 Les contraintes du routage des données dans les réseaux de capteurs

Les réseaux de capteurs [13] sont des systèmes sans fil à commutation de paquets. Ils partagent de nombreuses propriétés avec les autres réseaux, et sont particulièrement similaires à MANET dans de nombreux aspects (par exemple, l'auto-organisation, les sauts multiples et l'absence d'une infrastructure). Le choix d'une stratégie de routage pour un réseau de capteurs dépend de plusieurs contraintes comme l'environnement de propagation, la capacité des nœuds et les exigences de l'application de détection. Afin de créer un bon cadre d'intégration des applications, la conception efficace d'un réseau de capteurs est liée à plusieurs paramètres tels que :

Livraison en temps réel : certaines applications nécessitent que le message soit livré dans un délai bien déterminé, sinon le message devient inutile après la limite de temps. Par conséquent,

L'objectif principal de ces protocoles est de contrôler complètement la latence du réseau. Les performances des protocoles de routage peuvent être évaluées en contrôlant.

Durée de vie du réseau : l'application doit s'exécuter au niveau des nœuds de capteurs aussi longtemps que possible. Par conséquent, les protocoles visant cette préoccupation tentent d'équilibrer la consommation d'énergie de manière égale entre les nœuds en prenant en compte leurs niveaux d'énergie résiduelle. Cependant, la métrique utilisée pour déterminer la durée de vie du réseau dépend aussi de l'application. La plupart des protocoles considèrent que la durée de vie du réseau est la période de temps jusqu'à ce que le premier nœud meure.

Trouver un chemin entre un nœud source et sa destination se heurte à un certain nombre de difficultés dues à l'obligation d'optimiser la consommation d'énergie, respecter les capacités faibles des nœuds capteurs et assurer une bonne qualité de services. Ces difficultés sont causées par le support de communication sans fil utilisé, la taille du réseau, le déploiement, souvent aléatoire des nœuds, la dynamique du réseau et du passage à l'échelle.

3 Protocole de diffusion de données basé sur un arbre orienté puits pour les RCSFs à puits mobiles

De nombreux protocoles de diffusion de données et de routage sont proposés pour résoudre le problème de surcharge du réseau et pour minimiser le coût de communication causé par la mobilité puits. La plupart de ces approches sont hiérarchiques et peuvent être classées en trois catégories : grille, arbre, dorsale. A titre d'exemple, CMR (Coordinate Magnetic Routing) [14], TTDD (two-Tier Data Dissemination in Large-scale Wireless Sensor Networks) [15] et GBEER (Grid-Based Energy-Efficient Routing) [16] sont basés sur la construction d'une grille virtuelle pour gérer la diffusion des données et le routage vers les puits. CMR est un protocole de diffusion de données efficace pour les puits mobiles, il utilise le concept de coordination et de champ magnétique pour gérer la diffusion des données. Le TTDD utilise une structure hiérarchique de grille virtuelle. Néanmoins, TTDD construit une grille virtuelle par nœud source (c'est-à-dire orientée source) pour couvrir l'ensemble du réseau. GBEER est un protocole basé sur la grille mais il construit une grille unique après avoir déployé les nœuds.

Le protocole proposé dans [17], STDD (Sink-oriented Tree based Data Dissemination), combine les avantages des protocoles orientés source et des protocoles à structure unique. Il se déroule en trois phases : phase de construction, phase de mise à jour et phase de maintenance. Nous expliquons d'abord le modèle proposé par les définitions 1 et 2 puis expliquer les trois phases du protocole STDD à travers l'exemple de **la Figure 5**.

Définition-1: séquence de traces du puits : nous définissons un ensemble de puits mobiles comme $B_m = \{b_i/m \geq i \geq 0 \text{ (entier)}\}$ et définissons un ensemble de nœuds de capteurs statiques $S_n = \{S_j \mid n \geq j \geq 0 \text{ (entire)}\}$. Lorsque les puits b_i se déplacent dans la portée du nœud S_j , le nœud S_j est appelé le nœud d'accès pour b_i . La séquence de traces du mouvement du puits est définie comme un multi-ensemble de nœuds d'accès tel que $\gamma(b_i) = \{S_h \rightarrow S_v \rightarrow \dots \rightarrow S_p \rightarrow S_c\}$. Cette séquence représente les nœuds d'accès, dans lesquels b_i entre et sort séquentiellement de leurs plages. Lorsque b_i quitte S_h pour S_v , S_h est appelé le nœud d'accès précédent tandis que S_v est appelé le nœud d'accès actuel. Le premier nœud d'accès S_h dans le multi-ensemble $\gamma(b_i)$ est appelé le nœud d'accès de tête, tandis que le dernier nœud S_c est appelé le nœud d'accès courant.

Définition-2: les arbres de diffusion sont définis comme un multi-ensemble $T_m = \{t_i\}$ i est l'indice de bi , c'est-à-dire que chaque puits mobile $bi \in T_m$ a un arbre de diffusion unique $t_i \in T_m$. L'arbre de diffusion t_i , du puits mobile bi est un arbre couvrant qui couvre tous les nœuds dans S_n . Pour un $\gamma(bi) = \{Sh \rightarrow Sv \rightarrow \dots Sp \rightarrow Sc\}$ donné, l'arbre de diffusion t_i est construit par le nœud d'accès principal Sh (désigné par $t_i(Sh)$), c'est-à-dire que Sh est le nœud d'accès principale pour le puits bi et, en même temps, Sh est la racine principale de l'arbre de diffusion t_i . Le $t_i(Sh)$ sera mis à jour et maintenu lorsque le nœud d'accès de bi est changé en Sv (c'est à dire que bi quitte Sh pour Sv). cela signifie que t_i aura une nouvelle version $t_i(sv)$ et qu'il ne sera pas nécessaire de reconstruire un nouveau t_i . En d'autre termes, Sv est le nœud d'accès actuel de bi et l'arbre de diffusion t_i aura une racine à Sv . Par conséquence, il y aura des versions finies $\gamma(t_i) = \{t_i(Sh) \rightarrow \dots t_i(Sv) \rightarrow t_i(Sp) \rightarrow t_i(Sc)\}$ de t_i selon la fonction d'injection $f: \gamma(t_i) \rightarrow \gamma(bi)$. Soit $\alpha = |\gamma(t_i)| = |\gamma(bi)|$. C'est-à-dire le nombre de version de t_i dans $\gamma(t_i)$ ou le nombre de nœuds d'accès dans la séquence de traces du puits mobile $\gamma(bi)$. Soit $v_x^i (0 \leq x \leq \alpha)$ la version x de l'arbre de diffusion t_i dans $\gamma(t_i)$, c'est-à-dire $V_0 = t_i(sh); V_1 = t_i(sv) \dots V_{x-1} = t_i(sc)$, Pour un $\gamma(t_i)$ donné, la version $v_j^i (1 \leq j \leq \alpha)$ est mise à jour et maintenue sur la base de la version v_{j-1}^i . [17]

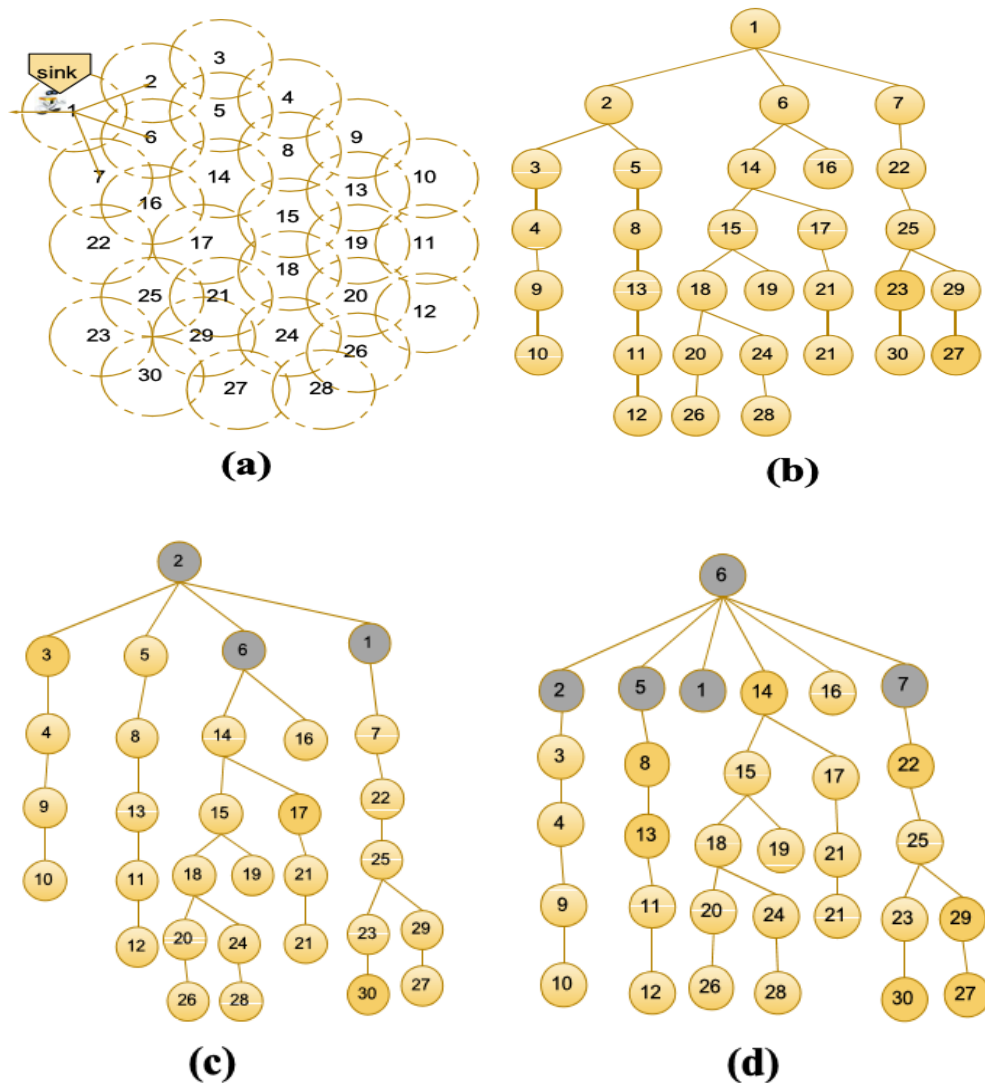


Figure 5 Un exemple de STTD.

3.1 La construction de l'arbre

L'arbre de diffusion est construit dans cette phase. Cette phase est exécutée une fois après la sélection du nœud d'accès principal. L'arbre de diffusion est un arbre couvrant qui s'étend pour couvrir tous les nœuds du réseau. Pour construire cet arbre, deux types de paquets sont utilisés : BLT_PARENT (l'expéditeur demande à être le parent du récepteur) et BLT_CHILD (l'expéditeur accepte d'être l'enfant du récepteur). De plus, chaque nœud a une liste de NODEPARENT (un parent par récepteur) et une liste de NODECHILD, comme le montre la Figure 5, de plus, chaque nœud possède une table pour stocker séparément les parents et les

enfants par puits, de telle sorte que chaque parent est une instance de l'objet NODEPARENT et chaque enfant est une instance de l'objet NODECHILD.

3.2 La mise à jour de l'arborescence

Pour un $\gamma(bi) = \{\dots Sp \rightarrow Sc\}$ et un $\gamma(ti) = \{\dots ti(Sp) \rightarrow ti(Sc)\}$ donnés, la version $V^j(1 \leq j \leq \alpha)$ est mise à jour sur la base de la version V^{j-1} lorsque bi quitte Sp pour Sc . Le processus de mise à jours a pour but de créer une nouvelle version de l'arbre de diffusion $ti(sc)$ en mettant à jour la version précédente $ti(sp)$ plutôt que de construire un nouvel arbre de diffusion. Pour ce faire, soient $NB(sp)$ et $NB(sc)$ être la liste des voisins de sp et la liste des voisin de sc , respectivement. De plus, soit $NB(Sc, Sp) = NB(Sc) \cap NB(Sp)$ l'intersection entre les deux listes. Le processus de mise à jour se déroule comme suit :

- (1) définir Sc comme étant la racine (c'est-à-dire $ti = (Sc)$) de l'arbre de diffusion ti .
- (2) tous les nœuds de $NB(Sc, Sp) \cup (NB(Sc)/NB(Sc, Sp))$ sont définis comme étant les enfants de Sc (y compris Sc).
- (3) les autres nœuds de $NB(Sp)/NB(Sc)$ sont définis comme enfants de Sp . Le paquet BLT_UPDATE est utilisé pour demander le processus de mise à jour de l'arbre de diffusion. Lorsqu'un nœud est sélectionné comme nouveau nœud d'accès pour l'arbre de diffusion, il envoie un BLT_UPDATE à tous ses voisins (c'est-à-dire à l'exclusion des nœuds qui figurent dans sa liste d'enfants).

Lorsque BLT_UPDATE est reçu par un nœud voisin : (1) si le nœud récepteur n'a pas de NODEPARENT pour le puits (c'est-à-dire le nœud d'accès précédent), il crée alors un nouveau NODEPARENT. Le nouveau nœud d'accès sera le parent du nœud d'accès précédent. (2) si le nœud récepteur a un NODEPARENT, il exécute deux opérations : d'abord, le nouveau nœud d'accès sera le parent du nœud récepteur. Deuxièmement, le récepteur envoie un paquet de notification à son ancien parent en lui demandant de se retirer (le récepteur sera supprimé). Parent qui demande à être retiré (le récepteur sera retiré de la liste des enfants de l'ancien parent).

BLT-CHILD	BLT-PARENT	BLT-UPDATE
SENDER	SENDER	SENDER
RECEIVER	RECEIVER	RECEIVER
SINK	SINK	SINK
	DTS	

NODEPARENT	NODECHILD	IS-SENT-FOR-SINK
PARENT	CHILD	IsSENT
SINK	SINK	SINK

Tableau 1 : Les paquets de contrôle et la structure du père et fils.

3.3 La maintenance de l'arbre

La première phase de STTD construit l'arbre de diffusion. Cet arbre sera mis à jour dans la deuxième phase au fur à mesure que le nœud d'accès est changé. Néanmoins, la phase de mise à jour ne garantit pas les sauts minimums et n'assure pas l'équilibrage du trafic, puisque la phase de mise à jour met à jour le premier niveau de l'arbre. Généralement, après avoir effectué la phase de mise à jour ou lorsque la valeur de α a augmenté pour $\gamma(b1) = \{...Sp \rightarrow Sc\}$ et $\gamma(t1) = \{...t1(Sp) \rightarrow t1(Sc)\}$, la surcharge des nœuds d'accès précédemment sélectionnés sera augmentée, puisqu'ils fonctionnent comme des routeurs pour les autres nœuds du réseau. De plus, le nombre de sauts sera augmenté pour certains nœuds.

La phase de maintenance a pour but de remédier à deux problèmes. Le premier est d'éviter l'augmentation du nombre de sauts (c'est-à-dire l'efficacité énergétique). Le deuxième objectif est d'améliorer l'équilibrage énergétique et d'éviter le problème des points chauds. En d'autres termes, les nœuds, qui ont été préalablement sélectionnés comme nœud d'accès, meurent plus rapidement que les autres nœuds du réseau. Ceci est similaire à GBEER, puisque les nœuds qui construisent la grille virtuelle peuvent mourir plus rapidement que les autres nœuds. Pour résoudre ces deux problèmes, soit $\gamma(b1) = \{...Sp \rightarrow Sc\}$ et $\gamma(t1) = \{...t1(Sp) \rightarrow t1(Sc)\}$. Après avoir exécuté la phase de mise à jour, seuls quelques nœuds en $t1$ souffrent de l'incrément du nombre de sauts. Par conséquent, nous pouvons classer les nœuds (chemins) en surs, non surs et équitables. Les chemins surs et équitables fonctionnent

correctement sans augmentation du nombre de sauts, tandis que les chemins non sûrs et équitables fonctionnent sans augmentation du nombre de sauts.

Pour gérer les chemins non sûrs. Pour un nœud donné $S_j \in (NB(S_p)/NB(S_p, S_c))$, si S_j chevauche avec un nœud $S_k \in NB(S_p, S_c)$, alors S_k sera le parent de S_j au lieu de S_p . Soit $S_x \in (NB(S_j)/NB(S_j, S_k))$, si S_x chevauche avec un nœud $S_y \in NB(S_j, S_k)$, alors S_y sera le parent de S_x . Cette sous-routine sera répétée de manière récursive aux nœuds feuilles du sous arbre dont la racine est S_j .

4 La congestion dans les réseaux de capteurs

Dans les réseaux sans fil, particulièrement dans les RCSFs, il existe plusieurs sources de congestion, comme le débordement des mémoires tampon, les transmissions concurrentes, la collision des paquets. Parmi les problèmes causés par la congestion, on peut citer : la perte de paquets qui contiennent parfois les informations critiques, le gaspillage de la bande passante [18].

4.1 Les types de congestion

La congestion est provoquée par les sources excédant la capacité (de stockage ou de traitement) des éléments du réseau. Pour cela, deux types de congestion pourraient se produire dans un RCSF :

4.1.1 La congestion au niveau du nœud (node-level congestion) :

Elle est répandue dans les réseaux conventionnels, et est provoquée par le débordement des tampons dans le nœud et peut causer la perte et le retard des paquets [19].

4.1.2 La congestion au niveau du lien (link-level congestion) :

Dans les RCSFs, le canal de communication sans fil est partagé par plusieurs nœuds en utilisant les protocoles de contrôle d'accès au média de la famille CSMA. Les collisions pourraient se produire quand les nœuds capteur essaient de transmettre en même temps suite à la détection d'un événement critique (incendie dans la forêt, séisme, etc.). Cette congestion conduit à un gaspillage de la bande passante et à un taux d'erreur élevé des paquets lors de leur réception [02].

4.2 Les étapes d'un mécanisme de contrôle de la congestion

Un mécanisme de contrôle de congestion passe en général par trois étapes : [20]

4.2.1 Détection de la congestion :

Les techniques classiques de détection de la congestion dépendent fortement de la perte de paquets due au dépassement de la mémoire tampon pour déduire la congestion et, dans une moindre mesure, de l'occupation des files d'attente et du délai de bout en bout. Comme l'énergie est la principale contrainte dans les WSN, les techniques traditionnelles ne peuvent être utilisées. Dans les WSN, la congestion peut être détectée de plusieurs façons, comme l'occupation de la mémoire tampon, l'échantillonnage du canal, taux de service des paquets et le taux d'ordonnancement.

4.2.2 Notification :

Une fois la congestion détectée, l'ensemble du réseau en est informé de l'une des deux manières suivantes :

- a) Notification explicite de congestion
- b) Notification implicite de congestion

En utilisant la notification explicite de congestion, un nœud congestionné notifie la congestion à l'utilisateur. Le reste du réseau en envoyant explicitement le paquet de contrôle. L'envoi de ce paquet de contrôle ajoute une charge importante à l'environnement déjà encombré. Par conséquent, la notification explicite de congestion n'a pas été utilisée par les protocoles de contrôle de congestion ultérieurs. Dans la méthode de notification implicite de congestion, l'information sur la congestion est diffusée au reste du réseau en écoutant les paquets de données qui sont en vol. Si une congestion est détectée, un bit de notification est ajouté à l'en-tête du paquet de données ou aux paquets d'accusé de réception (ACK) (s'ils sont utilisés). La notification implicite de congestion évite l'ajout de paquets supplémentaires au réseau lorsqu'il est déjà congestionné.

4.2.3 Prise de décision :

Que l'organe de décision se trouve du côté des machines d'extrémité, donc hors du réseau, ou dans le réseau maîtrisé par l'opérateur, le déclenchement de la décision de traiter la congestion se fait à l'examen des informations notifiées ou inférées du réseau et une fois ces informations jugées assez alarmantes pour devoir agir afin de prévenir une dégradation de la qualité de service [21].

4.3 Classification des protocoles de contrôle de la congestion

Nous pouvons différencier les protocoles de contrôle de congestion à travers plusieurs axes qu'on va décrire dans cette section [22].

4.3.1 Le mécanisme de détection de la congestion :

Le mécanisme de détection de la congestion peut être local ou global. La détection de congestion locale est réalisée par les nœuds intermédiaires en contrôlant des indicateurs locaux de congestion tels que l'occupation de la file d'attente ou l'état du canal. D'autre part, la détection de congestion globale est réalisée au niveau du puits ou les attributs de bout en bout tels que les retards inter-paquets et la fréquence de pertes peuvent être utilisés pour détecter la congestion.

4.3.2 L'objectif du contrôle de la congestion :

Dans leur nature, les RCSF sont orientés application. Donc, les protocoles de congestion seront différents selon l'application visée par le RCSF ou ils sont appliqués. Pour cette raison, les protocoles de contrôle de congestion sont aussi orientés application.

4.3.3 Les mécanismes de contrôle du taux de transfert :

Les mécanismes de contrôle du taux de transfert dans les RCSFs peuvent être centralisés, contrôlé par la source et par notification saut par saut. Le mécanisme de contrôle par la source est réalisé par le puits. Essentiellement, quand la congestion (ou le premier signe de congestion) est découvert, le puits donne l'ordre aux nœuds sources de régler leurs taux. Alors que, dans la notification saut par saut, le mécanisme est réalisé aux niveaux des nœuds intermédiaires, dans lesquels le nœud intermédiaire donne l'ordre aux nœuds qui sont en son amont de régler leurs taux en se basant sur son état de congestion local.

4.3.4 Equité et /ou Qos :

Classiquement, les protocoles de contrôle de la congestion sont chargés de réduire le taux de transmission afin d'éviter ou de réduire la congestion. En plus de cette tâche, d'autres exigences peuvent être envisagées. Cela inclut par exemple, les approches qui essaient de maintenir l'équité des flux opposés quand la congestion survient. De même, les approches QoS (Quality of Service) essaient d'allouer les ressources selon l'importance du flux ou les niveaux de réservation du canal.

4.3.5 Le modèle de l'application cible :

La plupart des protocoles se basent sur le modèle de communication plusieurs vers un. Cependant, quelques protocoles diffèrent sur leurs hypothèses dans ce modèle, telles que la Supposition du flux à haut débit (high-rate flows), ou multiples requêtes à multiples puits (multiple queries to multiple sinks), etc [23].

4.3.6 D'autres métriques :

D'autres métriques peuvent différencier les protocoles de contrôle de congestion. Par exemple, quelques protocoles nécessitent un support additionnel (MAC spécialisé, ou une capacité supplémentaire du réseau). De plus, quelques protocoles donnent une attention spéciale à l'efficacité énergétique.

5 Présentation de quelques protocoles de congestion

Dans les réseaux de capteurs sans fil, les mécanismes de contrôle de la congestion jouent un rôle important pour prolonger la durée de vie du réseau, améliorer l'équité et Qualité du service (QOS) en termes de débit, de taux de livraison des paquets et de distribution de bout en bout et aussi régler le problème des ressources limitées. Dans ce qui suit, nous présentons quelques mécanismes de contrôle de la congestion

5.1 Protocole SVM

Dans [24], les auteurs ont proposé d'utiliser la multi-classification obtenus via Support Vector Machines au niveau du nœud courant pour ajuster le débit de transmission dans le but remédier au problème de la congestion. Pour mieux expliquer leur solution nous présentons les paramètres définis comme suit :

5.1.1 Champ longueur de la file d'attente normalisée :

Le champ longueur de la file d'attente d'un nœud V définit le nombre de paquets dans le nœud V :

$$v_{q(v)} = \varphi(v)$$

La taille normalisée du tampon d'un nœud $V : v(\varphi(v))$ est définie comme le rapport d'occupation du tampon du nœud V :

$$B_r(v) = \frac{\text{nombre de paquets dans le tampon de file d'attente}}{\text{taille du tampon au nœud } V}$$

5.1.2 Niveau de congestion :

Parfois, la congestion se produit en raison d'un trafic intense. Par conséquent, la longueur de la file d'attente n'est pas la seule métrique pour reconnaître la situation du trafic dans chaque nœud. Alors, le Champ « Degré de congestion » C_d est défini. La valeur de C_d est définie comme suit :

$$C_d = \frac{T_s}{T_a}$$

T_a : Désigne l'intervalle de temps entre les arrivées de deux paquets de données adjacents dans la couche MAC.

T_s : Indique le temps de traitement moyen des paquets de données dans le nœud.

Si $C_d > 1$, le taux d'arrivée est supérieur au taux de départ de paquets de données. En d'autres termes, la congestion peut se produire dans un proche avenir dans ce nœud. Dans ce protocole, chaque nœud envoie les informations sur son trafic aux nœuds en son amont, puis la méthode de classification SVM est utilisée pour déterminer le taux de transmission exacte des données de chaque nœud en se basant sur l'information de charge de trafic envoyée par le nœud en aval et les deux paramètres définis précédemment B_r et C_d .

En effet, la classification basée sur les SVM dépend de ses paramètres. Par conséquent, chacune des machines à vecteurs de support doit être utilisée avec des paramètres appropriés. Dans ce protocole, ces derniers sont réglés simultanément pour toutes les machines à vecteurs

de support en utilisant l'algorithme génétique. L'algorithme génétique est mis en œuvre en utilisant le croisement uniforme et la mutation uniforme.

5.2 Protocole DAIPaS

Dans [25], un protocole léger d'évitement et de contrôle de la congestion, appelé Dynamic Alternative Path Selection Schème (DAIPaS) est proposé. Il est basé sur le contrôle de ressources au lieu de contrôler le taux de transmission des données.

En générale DAIPaS se compose de 4 étapes :

- Phase de configuration.
- Schéma de contrôle topologique.
- Soft Stage Schéma.
- Hard Stage Schéma.

5.2.1 Phase de configuration :

Elle est exécutée une fois au moment de l'initialisation du réseau pour établir les tables de voisinage. Cette phase est initiée par le nœud puits en diffusant un message HELLO avec son niveau qui est à zéro (le niveau indique la distance en sauts qui sépare un nœud et le puits). Chaque nœud qui reçoit ce message, répond avec un ACK, auquel le nœud puits répond par un message CONNECT. Ceci implique que nœud peut communiquer directement avec le puits, ce qui en fait de lui un nœud de niveau 1 et met à jour cette information dans la table de voisinage. Ce processus se répète au niveau de l'ensemble des nœuds du réseau, pour que chaque nœud établisse sa table de voisinage.

5.2.2 Schéma de contrôle topologique :

L'algorithme DAIPaS utilise un mécanisme dynamique de contrôle topologie sans ajouter toute charge supplémentaire au réseau. Il utilise les données des tables de voisinage qui ont été créées au cours de la phase de configuration et en particulier le niveau d'un nœud. Ainsi, chaque nœud transmet ses données à travers le nœud voisin dont le niveau est le plus petit (c'est le nœud le plus proche du puits). En conséquence un arbre dynamique est créé et chaque nœud transmet ses données par le chemin le plus court. Si un nœud a plus d'un nœud dans sa table voisinage au même niveau inférieure, il choisit le nœud émetteur de façon circulaire, après la fin de chaque session de transmission. En utilisant cette technologie, le trafic est réparti entre

ces nœuds conduisant ainsi à une consommation de l'énergie équilibrée et encore mieux à un partage équilibré de la charge.

5.2.3 Soft Stage Schème :

Les auteurs dans [25] notent que si les RCSFs sont constitués de nœuds avec des taux de transmission identiques, la congestion va se produire si plus d'un nœud transmet des données à un seul nœud relai et que leur débit de transmission cumulatif est supérieur au débit de données maximal que ce nœud. L'algorithme DAIPaS gère ce cas avec un schéma léger appelé Soft-Stage qui est décrit ici.

Lorsqu'un nœud reçoit des paquets de plus d'un flux, il tente d'arrêter la réception des données de plus d'un flux. Pour ce faire, il informe via un message, le nœud à partir duquel il préfère arrêter

De recevoir des données. Ainsi, il définit le champ « prochain numéro de séquence » dans l'en-tête du paquet ACK à « Faux » pour l'ID de nœud spécifique qu'il voudrait « Inhiber » sa transmission. Le nœud qui reçoit ce message doit rechercher un chemin alternatif. Pour des raisons de performance, chaque nœud choisit de continuer à envoyer avec le débit de données plus élevé. Si les nœuds « Inhibés » ne peuvent pas trouver un chemin alternatif approprié, ils continuent de transmettre des données à travers le même nœud en amont.

5.2.4 Hard Stage Schéma :

Si dans l'étape précédente un nœud choisit de garder son chemin initial mais ceci est inacceptable, l'algorithme « flag décision » est initiés. Ce dernier est un schéma très simple et son but est de changer la valeur du champ « Flag » dans la table de voisinage de vrai à Faux ou vice-versa lorsque les conditions sont appropriées. Le champ Flag indique si un nœud est disponible pour recevoir des paquets ou non. Chaque fois qu'un nœud modifie la valeur du champ flag, il informe immédiatement ses voisins à travers un paquet ACK modifié qu'il transmet. Un nœud peut devenir indisponible pour les raisons suivantes :

- L'occupation du tampon atteint sa limite supérieure.
- L'énergie résiduelle atteint sa limite inférieure.
- Indisponibilité d'un nœud de niveau inférieur.

5.2.5 Création d'un chemin alternatif :

Comme l'algorithme est dynamique, le nombre de sauts qui séparent un nœud et le puits peut changer lorsque l'état des nœuds à un niveau plus proche du puits change. Le choix d'un relai pour transmettre les données, après avoir évité un nœud congestionné, dépend d'abord de sa disponibilité (Flag) et le nombre de sauts le séparant avec le puits. Ainsi chaque nœud peut, d'une manière simple et facile, trouver le prochain nœud avec un calcul minimum. Il trie simplement le nombre de nœuds disponibles dans l'ordre croissant en fonction de leur nombre de sauts vers le puits et transférer ses données à travers le premier nœud de la liste. Si le premier nœud devient indisponible, le nœud émetteur choisit directement le nœud suivant dans la liste. Lorsque plusieurs nœuds sont au même niveau (même nombre de sauts), la table est triée en fonction de l'énergie résiduelle (au-dessus ou au-dessous de certains seuils spécifiques). Enfin, lorsque plus d'un nœud sont au-dessus de ces seuils, les nœuds sont triés en fonction du taux d'occupation du buffer. Dans le cas extrême où même cette valeur est la même pour plus d'un nœuds, l'algorithme choisit le nœud avec l'ID de nœud le plus petit comme relai. Selon l'application, une fonction pondérée pourrait être mise en œuvre, concernant le choix de la voie alternative, avec les quatre paramètres qui définissent le choix du chemin alternatif : Disponibilité (flag), nombre de saut, l'énergie résiduelle et le taux d'occupation du buffer.

5.3 HRTC

HRTC [26] (Hybride Ressource Traffic Control) est un protocole de contrôle de congestion hybride, il combine le contrôle des ressources et le trafic, dans les RSCFs. Pendant la congestion, un nœud congestionné informe un nœud source en utilisant une communication saut-par-saut pour réduire son débit de données. Lorsque ce message traverse les nœuds en aval en direction du nœud source, le protocole HRTC examine si le contrôle des ressources technique peut être appliqué aux nœuds traversés avant d'atteindre le nœud source. Si tel est le cas, HRTC abandonne la transmission du message. Si HRTC est incapable de trouver une autre route, le message continue son voyage vers sa destination, c'est-à-dire, le nœud source. Une fois que ce message atteint sa destination, il applique

Ensuite la technique de contrôle de trafic et modifie le taux de transmission du nœud source. De plus le nouveau débit de données est adapté par les nœuds du réseau.

5.4 Routage et control de congestion basé sur l'optimisation du taux de transmission

La méthode proposée dans [27] se concentre sur un nouveau mécanisme de control du taux de transmission basé sur le clustering et le routage par colonies de fourmis **Figure 6**. Au départ, les nœuds du réseau sont regroupés à l'aide d'un algorithme hybride K-means et Greedy best first search, puis, chaque nœud distribue sa distance et son énergie à travers tous les nœuds du cluster.

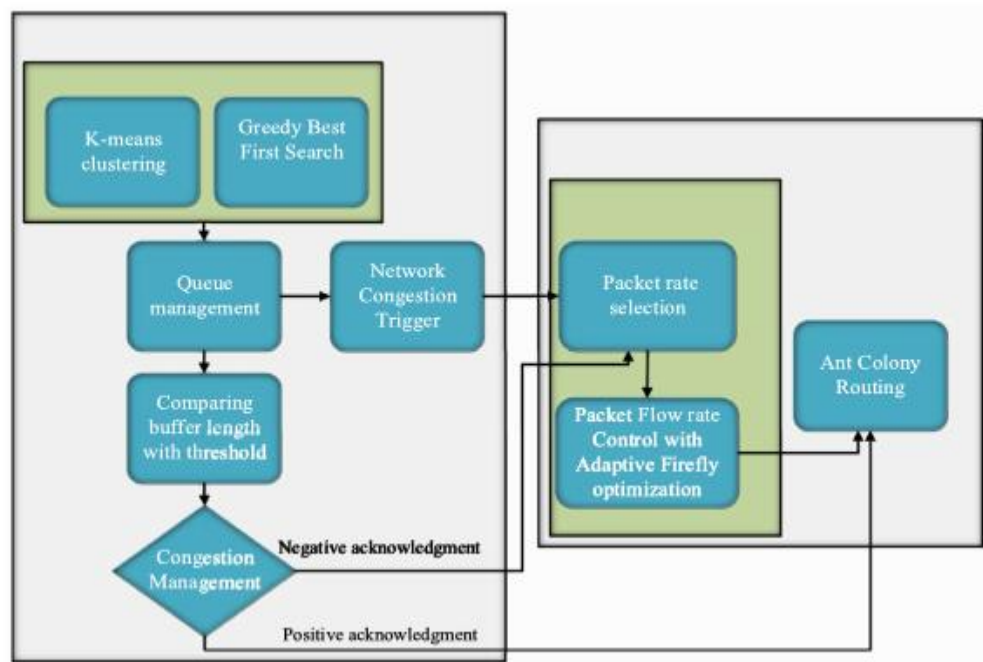


Figure 6 Représentation schématique de la méthodologie

La congestion du réseau est identifiée par le suivi du flux de paquets envoyés par de multiples sources vers le nœud puits. Dans cette solution, pour surveiller la congestion le paramètre utilisé est la longueur de la file d'attente. En effet, la taille de la longueur de la file d'attente est distribuée dans le réseau. Plus précisément, cette solution, lors de l'envoi d'un paquet d'une source à un autre nœud, il vérifie la taille de la mémoire tampon du nœud. Si cette dernière est surchargée, il recevra un accusé de réception nul. Alors, le taux de paquets sera minimisé par l'optimisation adaptative basée sur l'algorithme de luciole. Ci-après, les phases de la solution :

5.4.1 Phase de mise en grappes :

La transmission comprend l'envoi de paquets entre les nœuds et les têtes de cluster. Pour cela, nous introduisons l'algorithme K-means pour le regroupement des nœuds. Cet algorithme trouve d'abord le point central et estime la distance de Minkowski pour calculer la distance entre chaque nœud. A partir de là, le cluster sera formé.

5.4.2 Réduction du débit de paquets :

L'indication de congestion est envoyée via l'accusé de réception. Si l'accusé de réception est positif, il n'est pas nécessaire de minimiser le débit de paquets. Dans un autre cas, le taux de paquets est minimisé avec l'algorithme Adaptive Weight Firefly (AWF). L'objectif principal est de minimiser le taux de paquets envoyés.

Une fonction objective est mathématiquement formulée comme suit :

$$obj(i) = \min\{x\}, \text{ où } x \text{ désigne la valeur du débit de données.}$$

5.5 Algorithme de contrôle de la congestion basé sur l'algorithme d'optimisation multi-objectif (PSOGSA)

C'est un mécanisme basé sur l'algorithme d'optimisation multi-objectif [28], leur rôle consiste à l'optimisation la régulation du taux d'arrivée des données de chaque nœud enfant à nœud parent. La fonction d'optimisation multi-objectif prend en compte l'énergie de la fonction.

La transmission basée sur la priorité est activée comme l'approche d'optimisation, elle régule le taux d'arrivée en fonction de la bande passante et l'énergie du nœud enfant.

5.5.1 L'architecture du contrôle de la congestion à l'aide de PSOGSA :

L'architecture est représentée dans la **Figure 7**, quand le taux d'arrivée augmente considérablement plus que le taux de transmission, la congestion se produit. Si la congestion risque de se produire alors le niveau de la congestion est trouvé en utilisant la taille totale de la file d'attente. Le niveau de congestion correspond au stade auquel les paquets entrants commencent à s'accumuler dans la file.

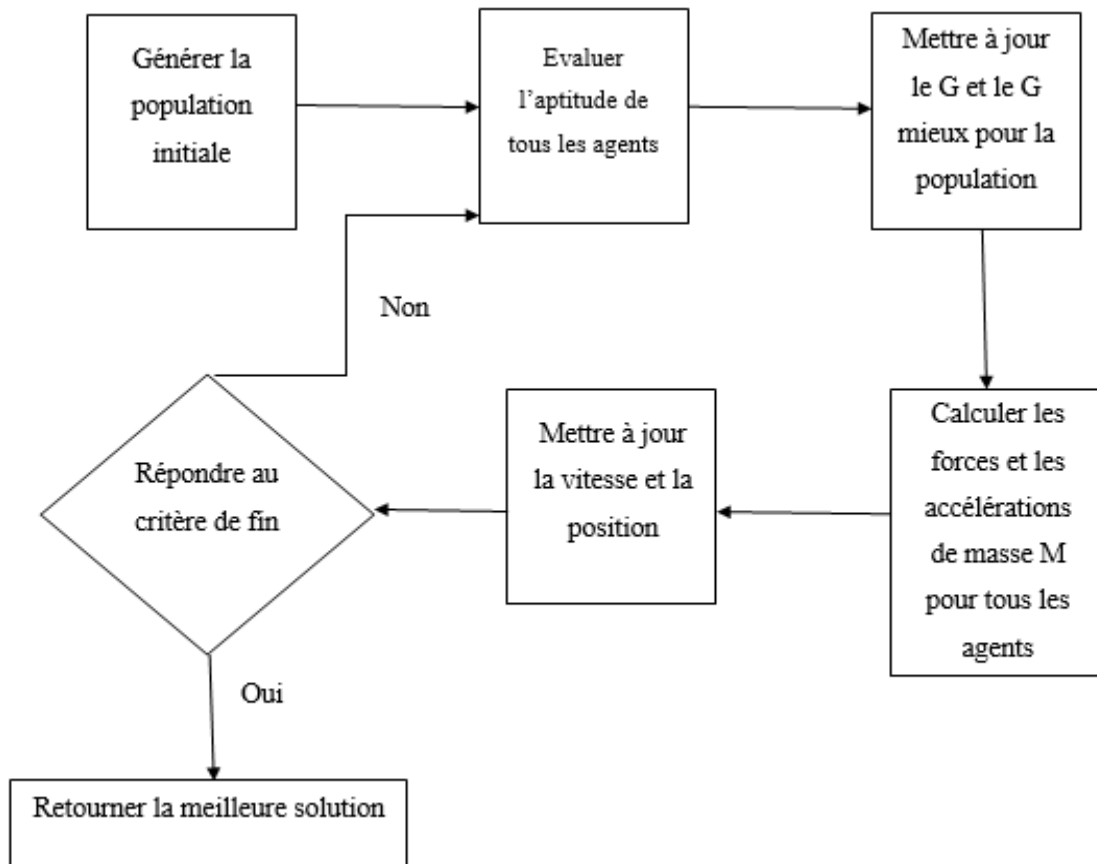


Figure 7 Architecture globale du schéma PSO-GSA proposé

5.5.2 Etat de congestion :

L'état de congestion est basé sur la valeur seuil choisi. La condition pour vérifier l'état de la congestion est donnée dans l'équation 1 :

$$\text{if } \sum_{n=1}^m v\varphi L_n > p\varphi \cdot \omega \text{ or } \forall l \in R, v\varphi L_l > 0,95 \cdot p\varphi \cdot CHP_l, \text{ where, } 0 < \omega \leq 1$$

$v\varphi L_n$: Désigne la taille du nième file d'attente virtuelle liée au nième nœud enfant.

CHP_l : C'est la priorité du nœud enfant.

$p\varphi$: Est la taille de la file d'attente physique.

5.5.3 Niveau de congestion :

Une fois la présence de la congestion dans le réseau est confirmée par l'état de la congestion, en suite on calcule le niveau global de congestion du nœud enfant par l'équation 2 :

$$\forall l, \text{if } \frac{v\phi Ln}{p\phi \cdot CHP_l} \geq 1 \text{ } Clevel_l = \frac{v\phi Ln}{p\phi \cdot CHP_l} - 1 \text{ OU } Clevel_l = 0$$

$p\phi \cdot CHP_l$: C'est la partie de la file d'attente physique (PQ) du dernier nœud enfant dans le nœud parent K dans [0,1]. Dans le nœud parent K dans [0,1].

$Clevel_l$: Est utilisé pour l'ajustement des taux.

5.5.4 Notification de congestion :

Le coût supplémentaire de la transmission des messages de contrôle peut être réduite en utilisant la notification implicite de congestion dans l'en-tête des paquets.

5.5.5 Optimisation des coûts basée sur PSO GSA pour le contrôle de congestion :

La congestion se produit au nœud parent l, parce que le taux de transmission du nœud parent et inférieure à celui du taux total des nœuds enfants. L'algorithme d'optimisation donne la fonction de condition physique basée sur les paramètres de congestion du réseau. La sélection du nouveau taux d'arrivée en utilise la valeur de la fonction de condition physique sélectionnée.

5.5.6 Solution de contrôle de congestion :

La solution de contrôle de congestion proposé trouve le nouveau taux d'arrivée. La réception simultanée de tous les nœuds enfants peut congestionnée nœud parent. A ce moment-là, la taille physique de la file d'attente est débordée, ce qui limite l'acceptation de paquets, conduisant à la perte de paquets. Pendant la retransmission, les paquets perdus sont tracés mais l'utilisation de la puissance du réseau est dépassée, entraînant une faible efficacité de réseau.

5.5.7 Fonction de fitness pour choisir la meilleure solution :

La fonction de fitness est utilisée pour l'optimisation du taux, plusieurs paramètres (Qos) sont pris en compte pour la maximisation de la fonction de condition physique. Les paramètres de la fonction d'aptitude proposée sont le taux d'arrivée des données disponibles,

la bande passante, la congestion, le débit et l'énergie. La fonction tient compte de ces six paramètres comme les plus importants pour contrôler la congestion du réseau. Et la fonction de fitness doit être maximale pour évaluation nouveau d'arrivée. La fonction de condition physique tenant compte de différents paramètres de qualité de service est donnée par l'équation suivante [28] :

$$f \rightarrow \max(f^{arrivalrate} + f^{bandwidth} + f^{trxrate} + f^{congestion} + f^{queueleth} + f^{energy})$$

6 Conclusion

Dans ce chapitre, nous avons présenté le problème de la congestion dans les RCSFs. Nous avons aussi présenté un état de l'art sur les protocoles de congestion proposés pour ces derniers. L'objectif de notre projet étant d'étudier la congestion dans les RCSFs à station de base mobile, nous avons choisi un protocole de routage proposé pour ces derniers à étudier et nous l'avons présenté dans ce chapitre pour étudier le niveau de surcharge des nœuds dans le chapitre suivant.

CHAPITRE 3

Proposition et la simulation

1 Introduction

Après avoir étudié le protocole de routage STDD et quelques protocoles contrôle de congestion proposés dans la littérature, nous analysons ici, dans ce chapitre, la congestion dans STDD pour proposer une solution pour éviter et corriger la congestion quand elle survient dans un réseau de capteur à station mobile.

2 Analyse de la congestion dans STDD

Dans le chapitre précédent, nous avons présenté un protocole de diffusion distribué axé sur les puits appelé Sink-oriented Tree based Data Dissemination (STDD) [17]. STDD construit seulement un arbre de diffusion principal pour chaque puits mobile. Lorsque le nœud d'accès est modifié, l'arborescence principale est mise à jour et maintenue pour garantir le chemin le plus court vers les puits mobiles. Nous avons remarqué que les auteurs de ce travail n'ont pas considéré le problème d'équilibrage de charge lors de la création et la maintenance des arbres de diffusion. En effet, des nœuds dans le réseau peuvent être utilisés plus que d'autres pour router les données collectées dans le réseau ce qui peut causer la surcharge de ces derniers et leur congestion. La congestion peut dégrader considérablement les performances des réseaux de capteurs. Elle réduit le taux de réussite de transmission puisque si un nœud est surchargé et n'a plus d'espace dans sa file d'attente, il supprime tous les messages reçus. Et elle cause une surconsommation de l'énergie des nœuds surchargés et réduit la durée de vie du réseau, puisque les messages supprimés sont retransmis par la source et l'énergie déjà consommé pour la transmission des messages supprimés est perdu pour rien. Si un message a avancé considérablement dans le réseau avant d'être supprimé i.e. Que le message est généré par un nœud très loin du nœud puits et le nœud surchargé est proche de ce dernier, le nombre nœuds ayant participé à son envoi est important et ils ont perdu leurs énergies dans la transmission d'un message qui n'est pas arrivé à destination et sera retransmis.

Pour analyser ce phénomène dans le protocole STDD, nous avons exécuté un scénario de simulation sur le simulateur que les auteurs ont développé pour leur travail et que nous avons récupéré du site [29].

3 Environnement de travail

Toolkit est une boîte à outil qui est développée pour simuler le protocole STDD. Cette boîte à outils est développée à l'aide de C # et WPF dans .NET4.5.

La figure présente l'interface du simulateur qui contient une interface graphique simple et facile pour l'utilisation.

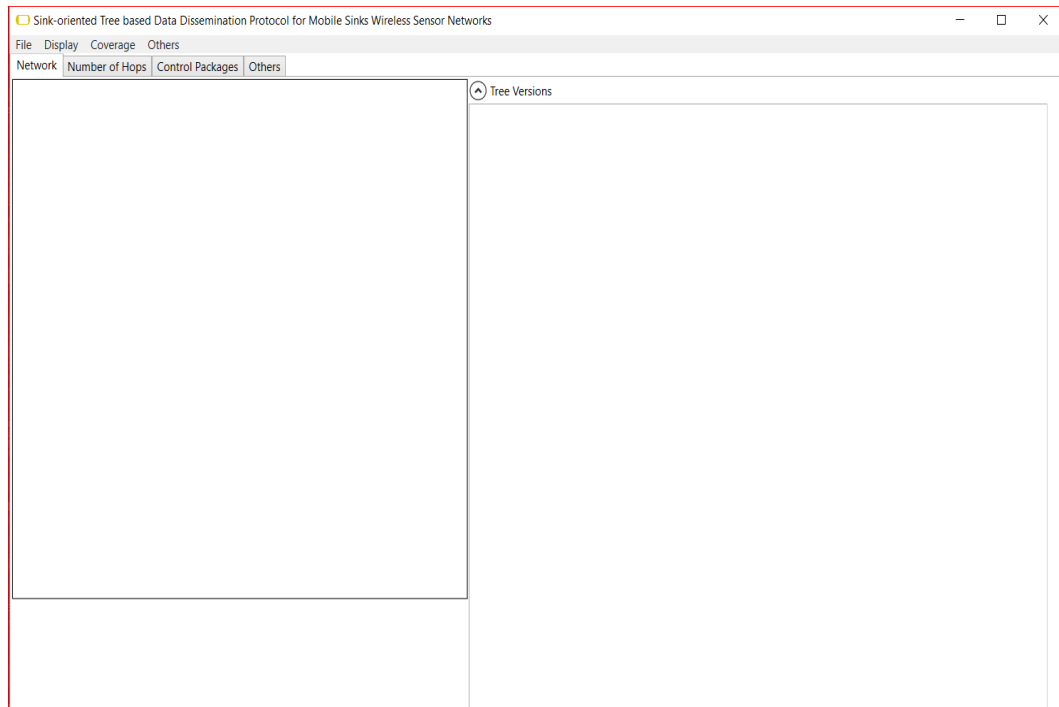



Figure 8 L'interface du simulateur.

4 Les étapes de la simulation

Pour analyser la congestion dans STTD, nous avons exécuté un scénario de simulation avec un ensemble des paramètres d'entrées qui sont indiqués dans les étapes suivantes :

- 1) En cliquant sur l'icône  dans le dossier "Toolkit", l'écran principal comme illustré à la **figure 8** ci-dessus s'ouvre.
- 2) Importer la topologie du réseau: "file" → "Import topology", plusieurs topologies comme le montre la **figure 9** sont affichées.
- 3) Nous avons sélectionné la topologie 1 qui est constituée de 100 nœuds puis cliqué sur "import". Le réseau de la topologie 1 est tel qu'il est présenté dans la **Figure 10**
- 4) Maintenant il faut sélectionner la couverture aléatoire avec le menu "Coverage" et en sélectionnant "Random" : "Coverage" → "Random"
- 5) Ajouter Sink mobiles : File → Add Mobile Sinks; une fenêtre apparaîtra comme illustré à la **figure 11**.
- 6) Nous avons choisi le nombre de puits à un seul, et sa vitesse à 20 m/s, comme illustré dans la **figure 12**. Le simulateur a choisi aléatoirement la position du nœud puits devant le nœud 38, comme on peut le voir dans la figure 5, le nœud puits est présenté par un 1 en rouge. Puis nous l'avons déplacé vers le nœud 39.

- 7) Ajouter les nœuds source : File→ Select Source Nodes Randomly. Une nouvelle fenêtre apparaîtra comme celle de la **figure 12**. On a choisi 12 nœuds source, avec un intervalle de génération de paquet à 1s et Intervalle de sélection de Source à 2s.
- 8) Enfin, nous avons démarré la simulation comme suit : Fichier-> Démarrer la simulation.

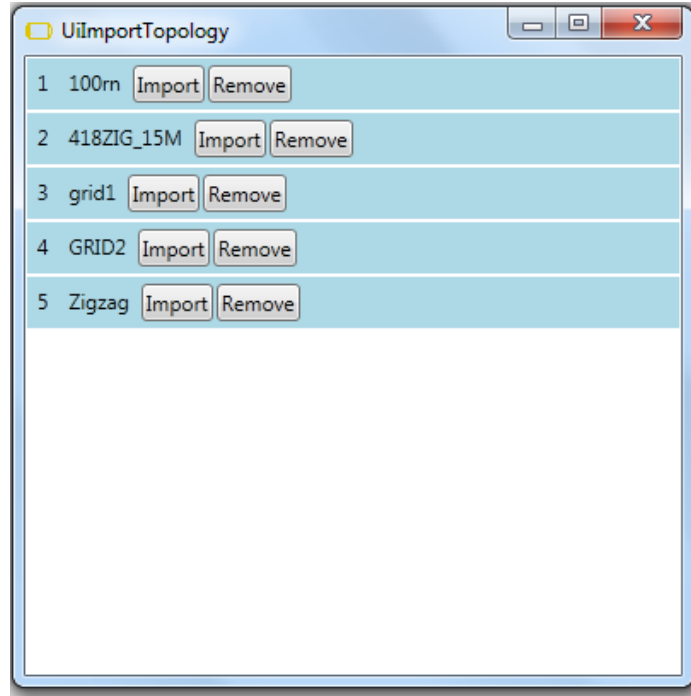


Figure 9 Importer la topologie

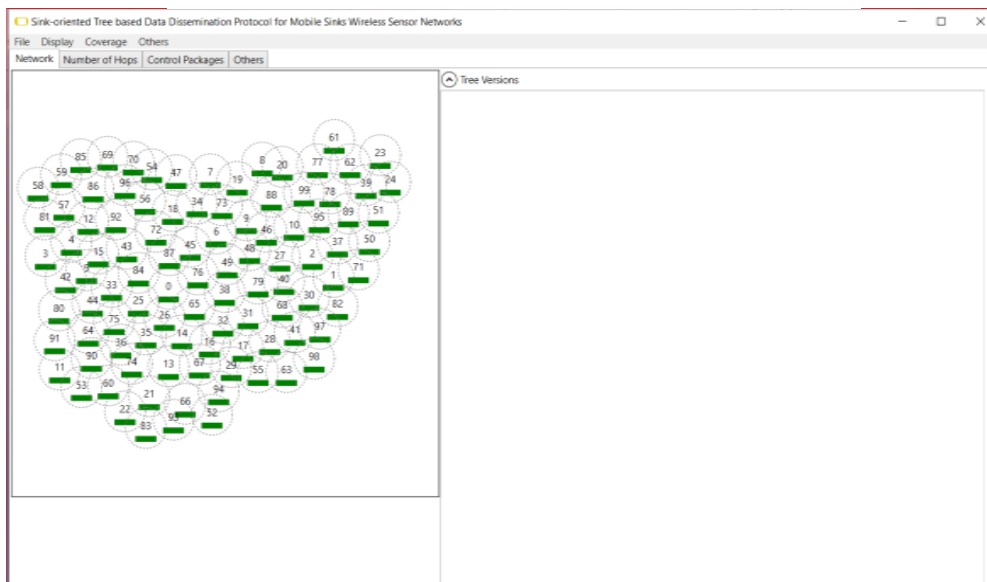


Figure 10 Le réseau de 100 nœuds déployés sur le terrain

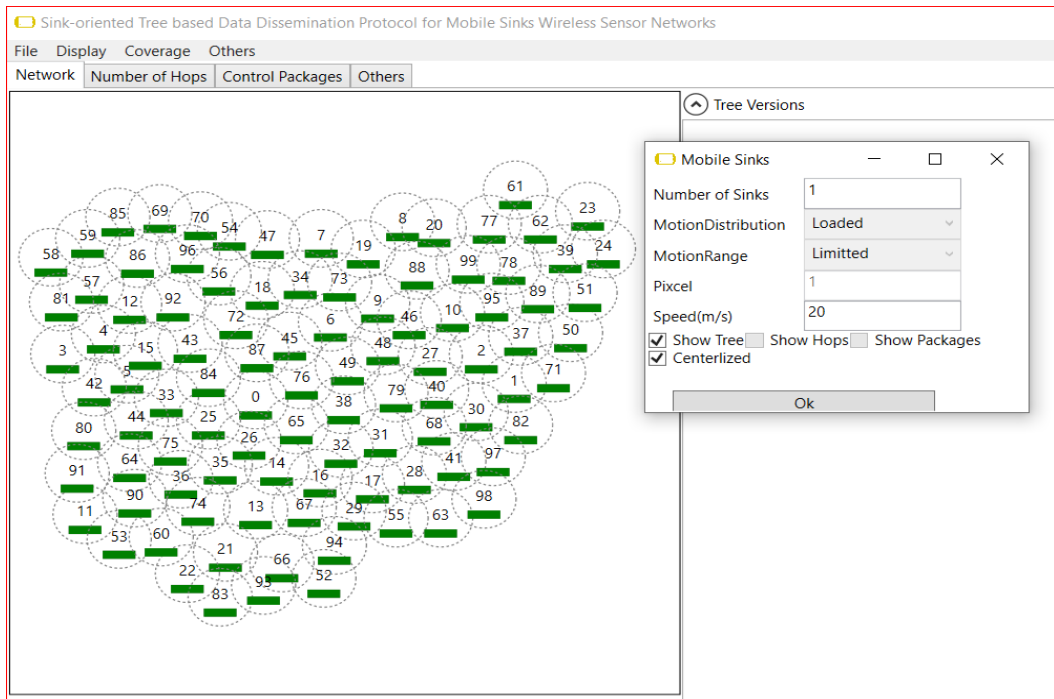


Figure 11 Les paramètres concernant le nœud puits

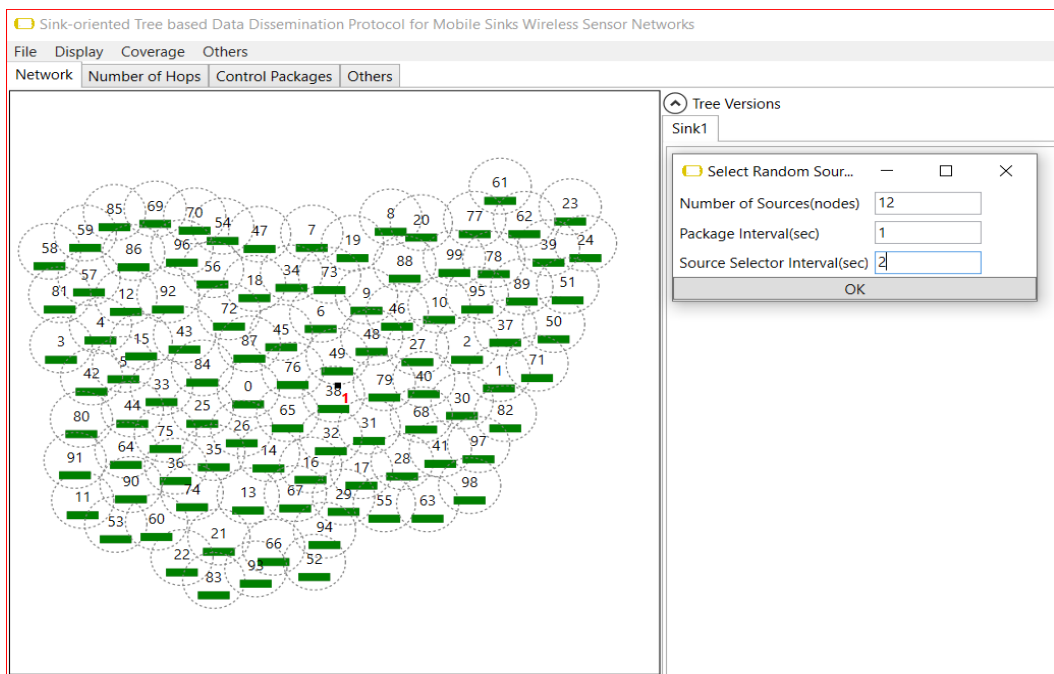


Figure 12 Les paramètres concernant les nœuds sources

5 Résultats de la simulation

Lors de la simulation la station mobile se trouve au niveau du nœud 39 et pour récolter les données des nœuds sources, STDD a créé un arbre de diffusion qu'on peut visualiser comme montré sur la **figure 13**. Le nœud puits se déplace et stationne sur le nœud 89 et mis à jour l'arbre

De diffusion vers cette nouvelle position comme montré dans la **figure 14**. Encore une fois, le nœud puits se déplace et se stationne au niveau du nœud 51 pour récolter les données des nœuds sources à travers la nouvelle version de l'arbre telle qu'elle est présentée dans la **figure 15**.

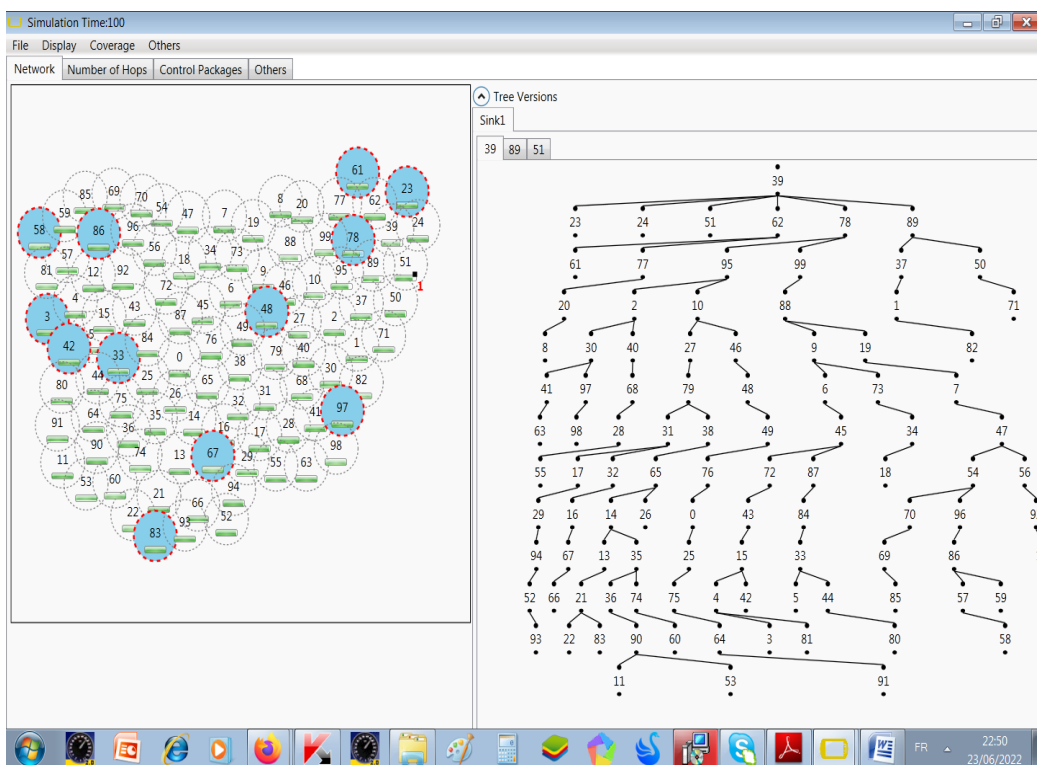


Figure 13 Première version de l'arbre pour la position du puits au niveau du nœud 39

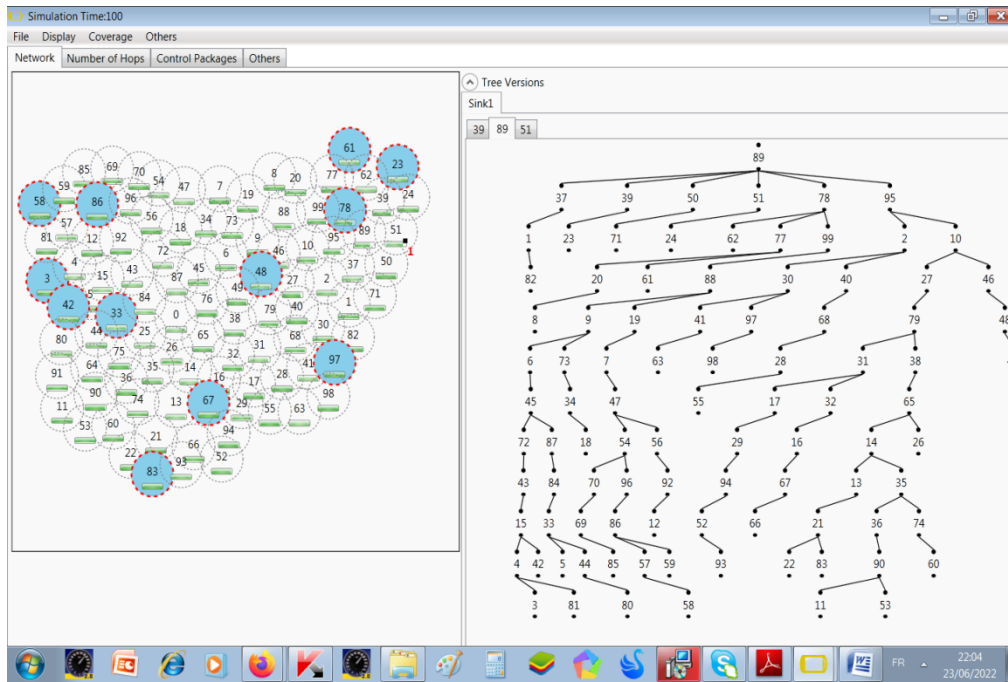


Figure 14 Deuxième version de l'arbre pour la position du puits au niveau du nœud 89

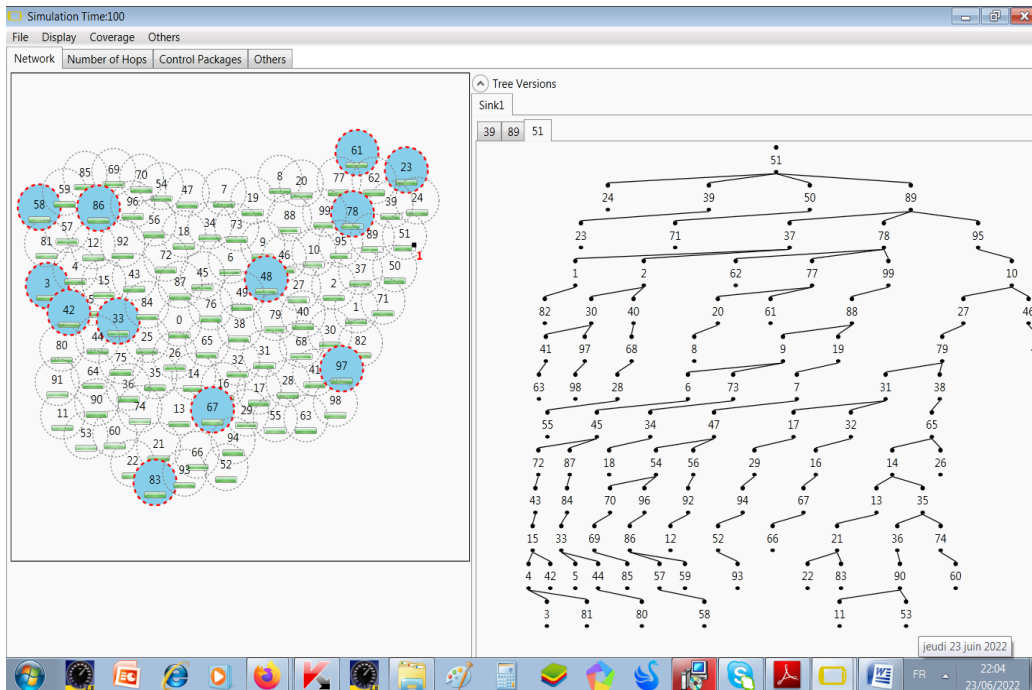


Figure 15 Troisième version de l'arbre pour la position du puits au niveau du nœud 51

Après un temps de simulation de 100s, le réseau ne fonctionne plus i.e. Qu'un nœud a épuisé complètement son énergie.

Nous avons analysé les résultats de simulation et nous avons remarqué que chaque arbre contient une racine qui est le nœud auprès duquel le nœud puits est stationné. Cette dernière collecte les données envoyées par tous les nœuds sources du réseau, et le déplacement du nœud puits induit la mise à jour de l'arbre et le changement de la racine. Ceci est censé équilibrer la charge des nœuds du réseau pour éviter la congestion, éliminer les pertes des paquets et équilibrer la consommation de l'énergie pour augmenter la durée de vie du réseau. Alors, L'idéal serait d'identifier dans chaque arbre les nœuds qui sont très sollicités et d'éviter de les surcharger dans le prochain arbre. Mais, nous avons remarqué que STDD ne fait pas ça ce qui résulte en la surcharge de quelques nœuds plus que d'autres dans les différents arbres et la congestion de ces nœuds causant ainsi des pertes de données, de l'énergie et de temps. Pour mieux comprendre ceci, nous avons présenté dans le **tableau-2** le pourcentage de nœuds pour lesquels un nœud relaie des données vers le nœud puits. Nous avons ensuite, établi des histogrammes pour mieux comprendre les résultats présentés dans le tableau. D'après ces résultats, nous avons remarqué que le nœud 39 est utilisé dans le premier arbre comme racine et route la totalité des paquets générés dans le réseau, mais après que le nœud puits s'est déplacé il a été allégé dans les deux autres arbres. Ceci va permettre au nœud 39 de terminer d'envoyer les paquets déjà reçu vers le nœud puits sans qu'il reçoit encore d'autre et ceci va éviter de le congestionner.

Par contre les nœuds 78, 89 et 95 sont des nœuds qui sont trop sollicités en permanence durant toute la simulation. Alors ce sont les nœuds les plus susceptibles d'être congestionnés. Par exemple le nœud 89 est la racine du deuxième arbre et dans le troisième arbre il est utilisé comme relai par plus que la moitié des nœuds du réseau.

Nœuds	taux d'envoi1	taux d'envoi2	taux d'envoi3
39	100	2	2
62	5	1	1
78	85	42	43
89	6	100	94
77	3	4	4
61	1	1	1
23	1	25	1
24	1	1	1
51	1	2	100
95	48	48	37
99	36	36	36
37	3	3	13
50	2	2	2
1	2	2	2
71	1	1	1
2	10	10	10
10	37	37	37

Tableau 2: Résultats de la simulation

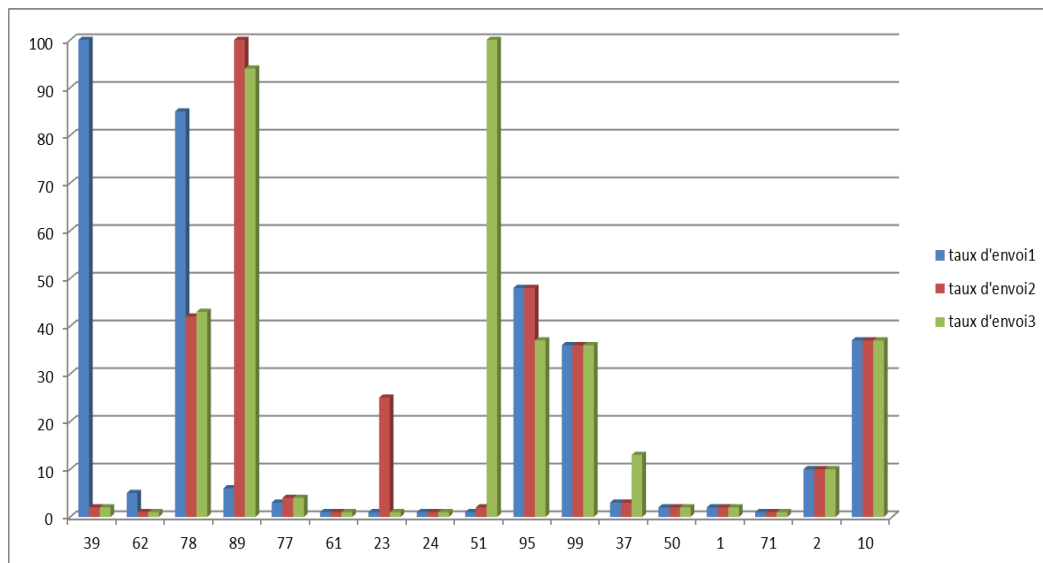


Figure 16 Résultat de simulation.

La **figure-16**, montre l’existence de la congestion au niveau de quelques nœuds, alors que d’autres ne sont pas vraiment utilisés comme 62, 71, 50, 37, 1, 61, 77 et 24. Pour les nœuds 62, 61, 24 et 77, le fait qu’ils ne sont pas utilisés est justifié puisque c’est des nœuds qui sont à l’extrémité du réseau, mais pour les autres, c’est un problème d’équilibrage lors du calcul des arbres. Pour les nœuds 39 et 51, ils sont désignés comme racine de l’arbre dans l’une des versions

mais grâce au nœud puits mobile qui se déplace, ils sont complètement allégés dans les deux autres versions de l'arbre. Par contre au niveau des nœuds (78, 89,99, 95) on a remarqué une surcharge considérable pour les trois versions des arbres.

6 La solution proposée pour le contrôle de la congestion dans STDD

Pour résoudre le problème de la congestion qui survient dans le réseau de capteurs lors de l'utilisation du protocole de routage STDD, nous proposons une solution hybride basée sur le contrôle des ressources et du trafic qui se base sur l'algorithme de lucioles ou Firefly en anglais. Notre solution vise à optimiser la conception des arbres de diffusion pour régler le problème de l'équilibrage remarqué dans le protocole pour éviter de surcharger quelques nœuds plus que d'autres, puis si des nœuds restent surchargés malgré un bon équilibrage de charge, l'algorithme va calculer le taux de génération de données optimal pour chaque nœud source pour réduire le flux arrivant à ces nœuds surchargés. Dans ce qui suit, nous présentons l'algorithme de lucioles.

7 Présentation algorithme de luciole

Les lucioles (en anglais Firefly) [30] sont de petits coléoptères ailés capables de produire une lumière clignotante froide pour une attraction mutuelle. Les femelles peuvent imiter les signaux lumineux d'autres espèces afin d'attirer des mâles qu'elles capturent et dévorent.

Les lucioles ont un mécanisme de type condensateur, qui se décharge lentement jusqu'à ce que certain seuil soit atteint. Ils libèrent l'énergie sous forme de lumière. Le phénomène se répète de façon cyclique.

7.1 Principe de base de l'algorithme luciole

L'algorithme de luciole [31], développé par Yang à l'université Cambridge en 2007 est inspiré par l'atténuation de la lumière sur la distance et l'attraction mutuelle des lucioles, plutôt que par le phénomène de la lumière clignotante des lucioles. L'algorithme considère ce que chaque luciole observe au point de sa position, en essayant de passer à une plus grande source de lumière, que la sienne. Cet algorithme prend en considération les trois règles suivantes [32] :

- Toutes les lucioles sont unisexes, ce qui fait l'attraction entre celles-ci n'est pas en fonction de leur sexe.
- L'attraction est proportionnelle à leurs luminosités, donc pour deux lucioles, la moins lumineuse se déplacera vers la plus lumineuse. Si aucune luciole n'est lumineuse qu'une luciole particulière, cette dernière se déplacera aléatoirement.

- La luminosité des lucioles est déterminée en fonction d'une fonction objective (à optimiser).

Le processus de l'algorithme de luciole commence avec l'initialisation de la population des lucioles et donc chaque luciole dans une population représente une solution candidate.

Taille de la population détermine le nombre de solutions ou la taille de l'espace de recherche dont le but est d'orienter la recherche à la meilleure localisation. Dans l'étape suivante, chaque luciole est évaluée en fonction de leur condition physique (intensité lumineuse). À chaque nouvelle étape itérative, la luminosité et l'attraction de chaque luciole est calculée.

La fonction de l'attractivité est définie en utilisant l'intensité lumineuse, la distance entre lucioles, et un coefficient d'absorption.

Le mouvement de luciole est défini par une fonction de mouvement, en utilisant la position actuelle, l'attractivité et une marche aléatoire, après avoir comparé la luminosité de chaque luciole avec toutes les autres lucioles, les positions des lucioles sont mises à jour en se basant sur les règles de connaissances sur les lucioles et sur leurs voisins.

Après le déplacement, la nouvelle luciole est évaluée et l'intensité de sa lumière est mise à jour. Pendant la boucle de comparaison en paire, la meilleure solution actuelle est la mise à jour d'une manière itérative. Le Processus de comparaison par pair est répété jusqu'à la satisfaction des critères de résiliation.

Dans l'algorithme des lucioles, il y a 5 points importants : [33]

Intensité de lumière : dans le cas le plus simple pour les problèmes de minimisation, la luminosité, ou l'intensité lumineuse d'une luciole à un endroit particulier peut être choisie comme : $I(x) \propto 1/f(x)$.

Attractivité : dans l'algorithme des lucioles, la principale forme de la fonction d'attractivité peut être n'importe quelle fonction monotone décroissante telle que la forme générale suivante.

$$\beta_{ij} = \beta_0 * e^{-y_{ij}^r}$$

Distance : la distance entre 2 lucioles i et j au x_i et x_j peut être la distance cartésienne comme suit:

$$r = \sqrt{\sum_k^d (x_{i,k} - x_{j,k})^2}$$

Mouvement : le déplacement d'une luciole i attirée par une plus lumineuse (attrayante) luciole j , est déterminé par :

$$x_i = (1 - \beta_{ij})x_i + \beta_{ij}x_j + \alpha(rand - \frac{1}{2})$$

Où le premier terme et le second sont dus à l'attraction. Le troisième terme est la randomisation. α est le paramètre aléatoire et peut être constant. "rand" est un générateur de nombre aléatoire uniformément distribuée dans $[0, 1]$.

Nombre de génération : la convergence vers la solution optimale globale n'est pas garantie dans tous les cas même si les expériences dénotent la grande performance de la méthode. De ce fait, il est fortement conseillé de doter l'algorithme d'une portée de sortie en définissant un nombre maximum d'itération.

7.2 Les avantages et les inconvénients de l'algorithme de luciole

Toutefois, un scénario dans le quel FA est très utile est lorsque la fonction objective doit être réduite au minimum au nombre de solutions multiples. Bien que cela ne soit pas totalement évident, il se trouve que FA s'organise automatiquement en essais secondaires qu'on peut trouver plusieurs solutions simultanément.

Le principal inconvénient de ces méthodes est qu'elles n'ont pas toujours trouvé la bonne solution puisque l'espace de recherche est limité par la solution initiale. [34].

8 Conclusion :

Dans cette partie, nous avons parlé sur les résultats de la simulation du protocole de routage de diffusion des données avec les puits mobile, effectuée sur le simulateur Sink-oriented Tree based Data Dissemination Protocol for mobile Sinks Wireless Sensor Networks. Ces résultats ont permis de voir que le protocole de routage n'équilibre pas la charge entre les nœuds du réseau quand il route les données à travers les arbres de diffusion établis pour chaque position du nœud puits. Ceci peut causer des congestions au niveau des nœuds surchargés et dégrader considérablement les performances du réseau. Pour remédier à ce problème, nous proposons d'utiliser l'algorithme de lucioles pour équilibrer la charge de nœuds du réseau pour ainsi éviter la congestion puis réduire le flux de données lorsqu'une congestion survient malgré l'équilibrage.

CONCLUSION GENERALE

Les réseaux de capteurs constituent un axe de recherche très riche et peuvent être appliqués dans plusieurs domaines différents. Cependant, il reste encore de nombreux problèmes à résoudre dans ce domaine afin de pouvoir les utiliser dans les conditions réelles. L'un des problèmes qu'on peut rencontrer dans ce genre de réseau est la congestion causée par la surcharge des files d'attente des nœuds. Cette congestion est liée à la nature des réseaux de capteurs, principalement leur mode de déploiement. Quand un réseau est congestionné, il y a un gaspillage des ressources, il est donc important de mettre en place des protocoles de contrôle de congestion pour assurer le bon fonctionnement de ces réseaux

Nous avons choisi dans ce mémoire d'étudier le contrôle de la congestion dans les RCSFs à station mobile fonctionnant avec le protocole de routage STDD. Nous avons analysé le critère de l'équilibrage de charge dans ce protocole lors du routage des données vers la station de base en suivant ses changements de positions. Nous avons constaté que STDD n'équilibre pas la charge des nœuds relais ce qui peut causer des congestions au niveau des nœuds trop sollicités. Pour améliorer le protocole STDD et éviter la congestion des nœuds relais, nous avons proposé d'utiliser un méta heuristique pour l'équilibrage de charge en mettant à jour les arbres lors du changement de la position de la station de base et pour la réduction du taux de génération des données par les nœuds source dans le cas où une congestion survient malgré l'équilibrage de charge.

Nous avons commencé à implémenter notre solution sur le simulateur de STDD mais malheureusement, nous avons rencontré beaucoup de problèmes suite auxquels nous n'avons pas pu évaluer notre solution. Alors l'évaluation de notre solution reste comme une perspective et suite à ce projet qui peut être effectué par d'autres étudiants.

BIBLIOGRAPHIE

[01] : Hadjaz Imane, « Conception et réalisation d'un réseau de capteurs sans fils application : Agriculture de précision », mémoire de master, Université Mouloud Mammeri de TIZI OUZOU, faculté de génie électrique et d'informatique, 2016.

[02] : Djebaili Yasmine, « Un protocole de routage tolérant aux pannes pour les RCSFs », Thèse, Université de BATNA 2, 20/06/2018.

[03] : SARI Mounya Amal, « Etude du RSSI pour l'estimation de la distance dans les réseaux de capteurs sans fil », mémoire de master, Université Abou Bakr Belkaid TLEMEN, 2016-2017.

[04] : Kaddouri Ahlam, Boudaoud Chahrazed, « routage géographique multi-chemin dans les réseaux de capteur sans fils », mémoire de master, université abdou baker belkaid TLEMEN, 2015-2016.

[05] : MESSAI Sarra, « Gestion de la Mobilité dans les Réseaux deCapteurs Sans Fil », thèse, Préparée dans le cadre d'une cotutelle entre l'Université Ferhat Abbas Sétif 1 et l'Université Claude Bernard Lyon1, Informatique,2019.

[06] : Mansour Farid, Feniche Farid « Etude comparative entre deux protocoles de routage dans les réseaux de capteur sans fil ' AODV' et 'DSR' », mémoire de master, université MOHAMED EL BACHIR EL IBRAHIMI B.B.A, 2020-2021.

[07] : Bouzidi Zeyneb, Benameur Amina, « Mise en place d'un réseau de capteurs sans fil pour l'irrigation intelligente », mémoire de master, Université ABOU BAKR BELKAID– TLEMEN, 2012.

[08] <https://www.rapport-gratuit.com/reseaux-de-capteurs-sans-fil-mobiles-a-element-mobile-rcsfem/?fbclid=IwAR3fjpWwArrhvg12iYIEcuoX5KXey31XSCNAIrOZKMDjgR5IsCTMNIDeXE>

[09] : Derdour Yasmine, « Conception et Développement d'un Système d'Exploration basé sur les Réseaux de Capteurs sans Fil avec Sink Mobile », thèse, université ORAN, 2015

[10] : Kaissari Soufiane, Conception d'un Réseau de Capteurs Sans Fil,mémoire , Université Mohammed V École normale supérieure d'enseignement technique - Rabat Département Génie électrique,2015

[11] : Abdesselan Abdelhalim, Belouatek Mohammed, « conception d'un algorithme de routage basée sur l'heuristique de recuit simulé pour les réseaux de capteur à grande échelle », ABOUBAKR BELKAÏD – TLEMEN, département informatique, 2012-2013.

BIBLIOGRAPHIE

- [12] : BENABDALLAH Karima, « Optimisation d'un protocole de routage AODV dans les Réseaux de capteur sans fil », mémoire de master, Université ABOUBAKR BELKAÏD – TLEMCEM, 2017
- [13] : Oussama Drissi, « Implémentation d'une stratégie de routage multi (niveau de données d'un réseau de capteurs sans fil dans le domaine ferroviaire », mémoire de master, Université du QUEBEC A TROIS-RIVIERES, Génie électrique, 2014.
- [14]: Shih-Hao Chang, M. Merabti; H.M. Mokhtar, « Coordinate Magnetic Routing for Mobile Sinks Wireless Sensor Networks », 21st International Conference of Advanced Information Networking and Applications Workshops (AINAW'07),21-23 May 2007
- [15]: Haiyun Luo, Fan Ye, jerry Cheng, Songwu Lu & Lixia Zhang, «TTDD: Two-tier data dissemination in large-scale wireless sensor networks », Journal international de recherche, janvier2005
- [16]: Kisuk Kweon, Hojin Ghim, Jaeyoung Hong, Hyunsoo Yoon, « Grid-Based Energy-Efficient Routing from Multiple Sources to Multiple Mobile Sinks in Wireless Sensor Networks », conférence, 2009 4th international Symposium on Wireless Pervasive Computing,11-13 february 2009 date
- [17]: Ammar Hawbani, Hassan Kuhlani, Xingfu Wang, Saleem Karmoshi, « Sink-oriented Tree based Data Dissemination Protocol for Mobile Sinks Wireless Sensor Networks », Journal of Mobile Communication, Computation and information. April 2017
- [18]: Akhilesh A.Waoo, Varsha Tiwari, « Review of Congestion Control Techniques in Wireless Sensor Network », Journal international de recherche, Volume-3, Issue-1, April 2018, Published date: 06 May 2018
- [19]: G.Srinivasan, S.Murugappan, «A Survery of congestion control techniques in wireless sensor networks», Journal of Information Technology and Knowledge Management July-December 2011.
- [20]: N.Thrimoorthy, T.Anuradha, « Congestion Avoidance Routing in Wireless Sensor Networks », Journal of Computer Applications · May 2016
- [21] : M.Andre Luc Beylot, «Nouveau paradigmes de controle de congestion dans réseau d'opérateur», Thèse Doctoratde l'université de Toulouse, 29 novembre 2016.
- [22] : Gaoua Lilia, « implémentation d'un mécanisme d'agrégation des données dans le protocole de routage LEACH », mémoire de master, université MOULOU MAMMARI DE TIZI OUZOU, 2010-2011.

BIBLIOGRAPHIE

- [23]: Majid Gholipour, Abolfazl Toroghi Haghghat, Mohammad Reza Meybodi, « Hop-by-Hop Congestion Avoidance in Wireless Sensor Networks Based on Genetic Support Vector Machine », Journal international de recherche , 25 October 2016, <http://dx.doi.org/10.1016/j.neucom.2016.10.035>
- [24]: Charalambos Sergiou, Vasos Vassiliou, Aristodemos Paphitis, « Congestion control in wireless sensor networks through dynamic alternative path selection », Journal international de recherche, 1 Octobre 2014, <http://dx.doi.org/10.1016/j.comnet.2014.10.007>
- [25]: Syed Roohullah Jan, Mian Ahmad Jan, Muhammad Usman, Muhammad Alam, « State-of-the-Art Congestion Control Protocols in WSN: A Survey », journal international de recherche EAI.EU, 26 mars 2018
- [26] : Vikas Srivastava, Sachin Tripathi, Karan Singh, Le Hoang Son, « Energy efficient optimized rate based congestion control routing in wireless sensor network », Journal de l'intelligence ambiante et de l'informatique humanisée, 19 Septembre 2019.
- [27] : <http://staff.ustc.edu.cn/~anmande/STTD/Toolkit.rar>
- [28] : Karishma Singh, Karan Singh, Le Hoang Son, Ahmed Aziz, « Congestion Control in Wireless Sensor Networks by Hybrid Multi-Objective Optimization Algorithm », Journal de l'intelligence ambiante et de l'informatique humanisée, 22 March 2018, <https://doi.org/10.1016/j.comnet.2018.03.023>
- [29] : TOUDJI Abdelkader, DLIMI Abdelah, « une approche hybride pour conception d'un protocole de routage multi-saut dans les réseaux de capteurs sans fil », mémoire de master, Université Ahmed Draia, Adrar Département MI, option réseaux et système intelligents, 2016-2017.
- [30] : Khaled Guerraiche, Mostefa Rahli, Abdelkader Zebila and Latifa Dekhici. « Series-Parallel Power System Optimization Using Firefly Algorithm », International Journal on Electrical Engineering and Informatics - Volume 7, Number 1, March 2015
- [31]: Xingshi Hi, « Firefly Algorithm : Recent Advances and Application », School of Science, Xi'an Polytechnic University, No. 10 Jinhua South Road, Xi'an 710048, P.R. China
- [33]: Latifa Dekhici, Rabeh Redjem, Khaled Belkadi, Abderrahman El Mhamedi. « Algorithme de Lucioles pour le Soins à Domicile », International conference on complex systems and logistic; ICOSYL 2018, Apr 2018, Paris, France.

BIBLIOGRAPHIE

[34] : BERROUACHEDI Abd Elhak , BOUAID Mohammed El Amine, « Contribution d'une approche hybride Firefly bio-inspirée pour le diagnostic des données médicales », mémoire de master, Université ABDELHAMID IBN BADIS-MOSTAGHENEM, département informatique, 2020-202.

Résumé

De nombreuses recherches ont été réalisées pour améliorer la capacité de faire router une grande quantité d'informations, à une station de base mobile dans un réseau de capteur, tout en assurant une longue période de vie du réseau.

La congestion est l'une des problématiques majeures d'un réseau de capteurs. L'une des conséquences de cette congestion est la dégradation des performances du réseau, ainsi qu'une perte de données et une surconsommation de l'énergie des nœuds.

Dans ce travail, nous avons proposé d'utiliser l'algorithme de Firefly pour équilibrer la charge des nœuds relais dans un réseau de capteur à station mobile fonctionnant avec le protocole de routage STDD, puis réduire le taux de génération de données par les nœuds sources. L'objectif de notre solution est d'éviter la congestion causée par la surcharge de certains nœuds relais et de la corriger lorsqu'elle survient inévitablement.

Abstract

Extensive research has been conducted to improve the capacity for routing a huge quantity of data to a mobile base station in a sensor network while ensuring a long network lifetime.

The congestion is one of the major problems in sensor network. One of the consequences of this congestion is the network performance degradation, as well as a data loss and an over-consumption of the nodes energy.

In this work, we proposed to use the Firefly algorithm to balance the charging of the relay nodes in a sensor network with mobile station functioning with routing protocol STDD, then, to reduce the rate of data generation by the nodes sources. The aim of our solution is to avoid the congestion caused by the overcharge of some relaying nodes and to correct it when it occurs unavoidably.

Table des matières

Remerciement

Dédicace

Liste des figures

Liste des tableaux

Liste d'abréviations

Introduction générale.....1

Chapitre 1 Généralités sur les réseaux de capteurs sans fil

1 Introduction..... 3

2 Capteur sans fil 3

2.1 Définition capteur 3

2.2 Composant d'un capteur sans fil 3

2.2.1 L'unité d'acquisition :..... 4

2.2.2 L'unité de traitement :..... 4

2.2.3 L'unité de communication :..... 4

2.2.4 Batterie :..... 4

3 Définition d'un RCSF ou WSN (Wireless Sensor Network)..... 5

4 Fonctionnement d'un réseau de capteurs 5

5 Caractéristique des RCSFs..... 6

5.1 Densité importante des nœuds 6

5.2 Topologie dynamique 6

5.3 Auto-organisation 6

5.4 Scalabilité..... 6

6 Les domaines d'application dans les réseaux de capteurs sans fil..... 7

6.1 Applications militaires 7

6.2 Applications de santé 7

6.3 Applications environnementales	7
6.4 La domotique	8
7 Réseaux de capteurs sans fil à éléments mobiles (RCSFs-EM)	8
7.1 Différentes formes de mobilité	9
7.1.1 Mobilité des nœuds capteurs :	9
7.1.2 Mobilité de l'événement :	9
7.1.3 Mobilité du nœud-puits :	9
7.2 Architecture d'un RCSF avec nœud-puits mobile	10
7.2.1 Nœuds capteurs sources :	10
7.2.2 Nœud-puits mobile :	10
7.2.3 Nœuds relais :	10
7.3 Avantage de l'utilisation d'un nœud-puits mobile.....	10
8 La pile protocolaire	11
8.1 La couche physique.....	11
8.2 La couche liaison de données.....	11
8.3 La couche réseau	12
8.4 La couche transport :	12
8.5 La couche application	12
8.6 Les plans de gestion	12
8.6.1 Plan de gestion de mobilité :	13
8.6.2 Plan de gestion d'énergie :	13
8.6.3 Plan de gestion de tâche :	13
9 La problématique	13
10 Conclusion	13
Chapitre 2 Les protocoles de routage et de congestion dans les réseaux de capteurs sans fil	
1 Introduction.....	15

2	Routage dans les réseaux de capteurs à station de base mobile.....	15
2.1	Définition du routage	15
2.2	Les contraintes du routage des données dans les réseaux de capteurs	16
3	Protocole de diffusion de données basé sur un arbre orienté puits pour les RCSFs à puits mobiles	17
3.1	La construction de l'arbre	19
3.2	La mise à jour de l'arborescence.....	20
3.3	La maintenance de l'arbre.....	21
4	La congestion dans les réseaux de capteurs	22
4.1	Les types de congestion	22
4.1.1	La congestion au niveau du nœud (node-level congestion) :	22
4.1.2	La congestion au niveau du lien (link-level congestion) :.....	22
4.2	Les étapes d'un mécanisme de contrôle de la congestion.....	23
4.2.1	Détection de la congestion :.....	23
4.2.2	Notification :.....	23
4.2.3	Prise de décision :.....	24
4.3	Classification des protocoles de contrôle de la congestion.....	24
4.3.1	Le mécanisme de détection de la congestion :	24
4.3.2	L'objectif du contrôle de la congestion :	24
4.3.3	Les mécanismes de contrôle du taux de transfert :	24
4.3.4	Equité et /ou Qos :	25
4.3.5	Le modèle de l'application cible :	25
4.3.6	D'autres métriques :.....	25
5	Présentation de quelques protocoles de congestion	25
5.1	Protocole SVM.....	25
5.1.1	Champ longueur de la file d'attente normalisée :.....	26
5.1.2	Niveau de congestion :	26

5.2	Protocole DAIPaS	27
5.2.1	Phase de configuration :	27
5.2.2	Schéma de contrôle topologique :	27
5.2.3	Soft Stage Schème :	28
5.2.4	Hard Stage Schéma :	28
5.2.5	Création d'un chemin alternatif :	29
5.3	HRTC	29
5.4	Routage et control de congestion basé sur l'optimisation du taux de transmission.....	30
5.4.1	Phase de mise en grappes :	31
5.4.2	Réduction du débit de paquets :	31
5.5	Algorithme de contrôle de la congestion basé sur l'algorithme d'optimisation multi-objectif (PSOGSA).....	31
5.5.1	L'architecture du contrôle de la congestion à l'aide de PSOGSA :	31
5.5.2	Etat de congestion :	32
5.5.3	Niveau de congestion :	33
5.5.4	Notification de congestion :	33
5.5.5	Optimisation des tarifs basée sur PSOGSA pour le contrôle de congestion :	33
5.5.6	Solution de contrôle de congestion :	33
5.5.7	Fonction de fitness pour choisir la meilleure solution :	33
6	Conclusion	34

Chapitre 3 Simulation

1	Introduction.....	36
2	Analyse de la congestion dans STDD.....	36
3	Environnement de travail	36
4	Les étapes de la simulation	37
5	Résultats de la simulation	40
6	La solution proposée pour le contrôle de la congestion dans STDD	44

7	Présentation algorithme de Firefly	44
7.1	Principe de base de l'algorithme Firefly	44
7.2	Les avantages et les inconvénients de l'algorithme de luciole	46
8	Conclusion :	46
	Conclusion générale	47

Bibliographie

Table de matières

Résumé