

République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid– Tlemcen

Faculté des Sciences

Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

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

Thème

**Routage basé sur les algorithmes génétiques dans les
réseaux de capteurs à grande échelle**

Réalisé par :

IMINE Youcef

KADDOUR Sidi Mohammed

Présenté le 31 Mai 2016 devant le jury composé de:

- *Mr BENMAMMAR Badr* *Président*
- *Mr LEHSAINI Mohamed* *Encadrant*
- *Mr MANA Mohammed* *Examineur*
- *Mr BEKARA Chakib* *Examineur*

Année universitaire:2015-2016

Remerciement

Nous tenons à exprimer toute notre reconnaissance à Mr Lehsaini Mohammed. Nous le remercions de nous avoir proposé ce sujet, de nous avoir encadré, orienté, aidé et conseillé tout au long de ce projet.

Nous remercions Mr Badr Benmamar, Mr Mana Mohammed et Monsieur Bekara Chakib pour avoir accepté d'examiner ce projet et d'avoir consacré une partie de leur temps à sa lecture.

Nous remercions les membres du binôme pour leur travail dur le long de la réalisation de ce projet, leurs patiences ainsi que leurs efforts.

Enfin nous remercions toutes les personnes qui nous ont aidés, encouragé et soutenus le long de ce semestre.

DEDICACES

Je remercie en primordiale mes parents qui ont cru en moi et qui m'ont soutenu pendant mon parcours, Aucune dédicace ne saurait être assez éloquente pour leur exprimer tous les sacrifices que n'ont cessé de me donner depuis ma naissance, mon éducation et ma formation.

A mes frères ;

A mes oncles et tantes,

Mes cousins et cousines et à toute la famille ;

A mes grands-parents pour toutes leurs prières intarissables ;

A mon binôme Sidi Mohammed Kaddour;

A tous mes amis ;

Je dédie ce mémoire

YOUCEF

DEDICACES

A mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance. J'espère qu'ils trouveront dans ce travail toute ma reconnaissance et tout mon amour.

A mes chers frères et sœurs ;

A Youcef Imine et Sarah Ammam

Et tous mes amis ;

A mes tantes et à mes oncles ;

A chaque cousin et cousine et toute la famille.

Je dédie ce mémoire

SIDI MOHAMMED

Table des matières

TABLE DES FIGURES	4
LISTE DES TABLEAUX	6
INTRODUCTION GENERALE.....	7
1. CHAPITRE I GENERALITES SUR LES RESEAUX DE CAPTEURS SANS FIL	9
1.1. INTRODUCTION.....	9
1.2. QU'EST-CE QU'UN CAPTEUR ?	9
1.2.1. Définition d'un nœud capteur	9
1.2.2. Catégories de capteurs.....	11
1.3. LES RESEAUX DE CAPTEURS SANS FIL.....	11
1.3.1. Caractéristiques des réseaux de capteurs sans fil	12
1.3.2. Domaines d'applications	13
a. Applications militaires	13
b. Découvertes de catastrophes naturelles	14
c. Détection d'intrusions	14
d. Applications métier.....	14
e. Surveillance médicale	14
f. Contrôle d'édifices.....	14
g. Applications environnementales	14
h. La domotique.....	15
1.3.3. Catégories de communications dans les RCSF.....	15
a. Scénario périodique.....	15
b. Selon la demande	16
c. Scénario orienté événement (event-driven).....	16
1.4. CONTRAINTES DE ROUTAGE DANS LES RCSF	17
1.5. CONCLUSION	18
2. CHAPITRE II LES PROTOCOLES DE ROUTAGES DANS LES RCSF	19
2.1. INTRODUCTION.....	19
2.2. CRITERES DE PERFORMANCE DES PROTOCOLES DE ROUTAGE DANS LES RCSF.....	19
2.3. TAXONOMIE DES PROTOCOLES DE ROUTAGE DANS RCSFS	20
2.3.1. Données centrées:	22
a. SPIN: "Sensor Protocol for Information via Negotiation"	22
b. Directed Diffusion	23
c. Energy-aware routing	24
2.3.2. Protocoles hiérarchiques :	25
a. LEACH	25

b.	PEGASIS.....	27
c.	TEEN.....	27
d.	CHIRON.....	29
e.	ETR " Energy-aware Tree Routing protocol".....	32
2.3.3.	<i>Les protocoles basés sur la localisation</i>	33
a.	Geographic adaptive fidelity (GAF).....	33
b.	Geographic and energy-aware routing (GEAR).....	34
2.3.4.	<i>Principaux protocoles de routage basés sur les heuristiques</i>	35
a.	CRP "Comprehensive Routing Protocol ".....	35
b.	ACALEACH:" Ant Colony clustering algorithm ".....	37
c.	MSRP:" Multi-SinkRoutingprotocol".....	37
d.	Ant colony multicast trees (ACMT).....	38
2.4.	CONCLUSION.....	39
3.	CHAPITRE III TECHNIQUES D'OPTIMISATIONS POUR LES RESEAUX : LES HEURISTIQUES.....	40
3.1.	INTRODUCTION.....	40
3.2.	CLASSES D'ALGORITHMES POUR LES PROBLEMES NP_DIFFICILES.....	40
3.2.1.	<i>Les algorithmes d'approximation</i>	40
3.2.2.	<i>Les algorithmiques heuristiques</i>	41
3.2.3.	<i>Les algorithmes probabilistes</i>	41
3.2.4.	<i>Les méta-heuristiques</i>	41
3.3.	EXEMPLE DE META-HEURISTIQUE.....	41
3.3.1.	<i>La colonie de fourmis</i>	42
3.3.2.	<i>La méthode Tabou</i>	44
3.3.3.	<i>Le recuit simulé</i>	48
3.3.4.	<i>Les algorithmes génétiques</i>	49
3.4.	CONCLUSION.....	55
4.	SCHEMAS DE ROUTAGE BASES SUR LES ALGORITHMES GENETIQUES.....	56
4.1.	INTRODUCTION.....	56
4.2.	ENVIRONNEMENT DU DEVELOPPEMENT OMNET++.....	56
4.2.1.	<i>Les modules</i>	57
4.2.2.	<i>Les canaux de communication(Channel)</i>	58
4.2.3.	<i>Les messages</i>	59
4.2.4.	<i>Les fichiers de descriptions"Ned File"</i>	59
4.3.	SCHEMA DE ROUTAGE BASE SUR LES ALGORITHMES GENETIQUES.....	60
4.3.1.	<i>La désignation de la population initiale</i>	60
4.3.2.	<i>La fonction objective "fitness"</i>	65
4.3.3.	<i>L'étape de croisement</i>	65
4.3.4.	<i>L'étape de sélection</i>	66

4.3.5.	<i>Critère d'arrêt</i>	66
4.3.6.	<i>Choix de la meilleure solution</i>	67
4.4.	IMPLEMENTATION SUR OMNET++.....	68
4.4.1.	<i>Le module station de base</i>	68
4.4.2.	<i>Le module capteur</i>	68
4.4.3.	<i>Définition des types de messages</i>	68
4.5.	FONCTIONNEMENT DU RESEAU	69
4.6.	MODELE ENERGETIQUE	71
4.7.	SIMULATION	71
4.7.1.	<i>Paramètres de simulation</i>	71
4.7.2.	<i>Résultats de la simulation</i>	72
a.	La version énergie.....	72
b.	Version distance	74
4.8.	DISCUSSION DES RESULTATS	75
4.9.	COMPARAISON AVEC D'AUTRES PROTOCOLES :	75
4.10.	AMELIORATIONS PROPOSEES :	76
4.10.1.	<i>Changement de la méthode de sélection</i> :.....	76
4.10.2.	<i>Routage avec ajustement de puissance de transmission</i>	78
4.11.	CONCLUSION	79
	CONCLUSION GENERALE	80
	REFERENCES BIBLIOGRAPHIQUES	81

Table des figures

Figure 1.1:Fonctionnement d'un capteur	9
Figure 1.2:structure d'un capteur	10
Figure 1.3:exemple de RCSF	11
Figure 1.4:collecte d'information à la demande	16
Figure 1.5:Scénario orienté évènement	16
Figure 2.1 : Etapes de fonctionnement du protocole SPIN	22
Figure 2.2:Transmission des données dans le protocole Directed diffusion.....	23
Figure 2.3:Illustration du clustering dans le protocole LEACH	26
Figure 2.4:Exemple d'une chaîne de nœuds dans le protocole PEGASIS	27
Figure 2.5:Organigramme représentatif du protocole TEEN.....	28
Figure 2.6:Phase de construction des groupes	29
Figure 2.7:Phase d'élection.....	30
Figure 2.8:Acheminement des données	32
Figure 2.9:Définition des niveaux	33
Figure 2.10:Exemple d'une table de routage du protocole CRP	37
Figure 2.11:Illustration du fonctionnement du protocole MSRP	38
Figure 3.1:Expérience du double pont binaire	42
Figure 3.2: Effet de coupure d'une piste de phéromones	42
Figure 3.3:Pseudocode de l'heuristique colonie de fourmi	43
Figure 3.4:Illustration d'une recherche locale	45
Figure 3.5: Types de mémoires de recherches	46
Figure 3.6:Comportement des mémoires de recherche.....	46
Figure 3.7:Organigramme d'une recherche Tabou	47
Figure 3.8:Organigramme représentatif du recuit simulé	48
Figure 3.9:Formes de codage	50
Figure 3.10:Croisement en un point	50
Figure 3.11:Croisement en deux points.....	51
Figure 3.12:Croisement uniforme	51
Figure 3.13:Croisement arithmétique.....	52
Figure 3.14:Illustration d'une mutation	52
Figure 3.15:Sélection par roulette	53
Figure 3.16:Sélection par rang	53

Figure 3.17:Sélection par tournoi.....	54
Figure 3.18:Organigramme d'un algorithme génétique	55
Figure 4.1:Structure d'un réseau sur Omnet++	57
Figure 4.2: Exemple d'un module Omnet++	57
Figure 4.3: Hiérarchie des objets sous Omnet++	59
Figure 4.4:Quelques méthodes utiles pour manipuler un message	59
Figure 4.5:Définition d'un réseau	60
Figure 4.6:Illustration du fonctionnement de find_Route()	61
Figure 4.7:Illustration du fonctionnement de find_Route()	61
Figure 4.8:Cas d'utilisation de find_Enhanced()	62
Figure 4.9:pseudo-code de la procédure find_Route	63
Figure 4.10:Illustration du fonctionnement de find_Enhanced()	64
Figure 4.11:pseudo-code de la fonction find_Enhanced	64
Figure 4.12: Etape de croisement	66
Figure 4.13:Organigramme de l'algorithme proposé	67
Figure 4.14: Phase de découverte.....	69
Figure 4.15:Phase d'établissement des routes.....	70
Figure 4.16:Acheminement des données	70
Figure 4.17:Résultats de simulation sur la topologie 1	73
Figure 4.18: Résultats de la simulation sur la topologie 2	73
Figure 4.19: Résultats de la simulation sur la topologie 3	74
Figure 4.20:Résultat de la version distance de l'algorithme appliqué sur la topologie 2	74
Figure 4.21:Comparaison de la durée de vie de notre algorithme LEACH et LEACH-C.....	76
Figure 4.22:Résultat de la simulation avec l'ancienne méthode de sélection	77
Figure 4.23:Résultat de la simulation avec la nouvelle méthode de sélection.....	77
Figure 4.24:résultats de la simulation d'un routage avec ajustement de puissance testé sur la topologie 1.....	79

Liste des tableaux

Tableau 4-1:Paramètres de la simulation	72
Tableau 4-2:Description des topologies utilisées.....	72
Tableau 4-3:Résultats d'exécution des deux méthodes de sélection dans 5 topologies différentes	78

Introduction générale

Le besoin de la rapidité des échanges d'informations et le développement des communications ont abouti à une progression massive de la technologie sans fil. La facilité d'utilisation de cette technologie l'a imposé comme étant un élément essentiel et effectif dans notre quotidien.

Par ailleurs, l'architecture ad-hoc représente l'une des implémentations les plus utilisées dans le monde de la technologie sans fil. Sa caractéristique d'auto-organisation et l'évolutivité de la topologie lui permettent de développer plusieurs infrastructures telles que les réseaux mobiles, les réseaux de capteurs et dernièrement l'internet des objets. Cependant, cette technologie est confrontée à de nouveaux problèmes, nous citons parmi eux les déconnexions fréquentes, les débits variables et la quantité d'énergie limitée des nœuds faisant partie du réseau.

Le routage dans les réseaux de capteurs sans fil (RCSF) est un véritable challenge en raison de plusieurs caractéristiques qui les distinguent des réseaux ad hoc. Premièrement, il est impossible de construire un système d'adressage global pour le déploiement d'un grand nombre de nœuds de capteurs. Par conséquent, des protocoles basés sur IP classiques ne peuvent pas être appliqués aux RCSF. Deuxièmement, les données détectées dans les différentes régions (sources) sont acheminées vers une seule destination appelé station de base ou bien SINK il s'agit donc d'une communication de type many-to-one. Troisièmement, le trafic de données pourra générer une redondance importante puisque plusieurs capteurs peuvent collecter les mêmes données dans une région. Par ailleurs, une telle redondance doit être gérée par les protocoles de routage afin d'améliorer l'efficacité énergétique et l'utilisation de la bande passante. Quatrièmement, les nœuds capteurs sont étroitement limités en termes de puissance de transmission, d'énergie, de capacité de traitement et de stockage donc ils nécessitent une gestion judicieuse des ressources.

Dans le cadre de notre projet de fin d'études, nous nous intéressons aux réseaux de capteurs et en particulier au routage dans les réseaux de capteurs à grande échelle. Dans cette optique nous avons visé à optimiser les performances de routage dans ce type de réseaux tout en tenant compte de sa spécificité. Pour cela, nous avons utilisé les heuristiques pour améliorer les performances de routage dans les réseaux de capteurs denses vu la limitation des protocoles

classiques dans la découverte de routes lors du passage à l'échelle en particulier. À cet effet, nous avons implémenté une démarche de routage basée sur les algorithmes génétiques qui minimise la consommation d'énergie et augmente la durée de vie des réseaux mis en place.

Ce mémoire est organisé en quatre chapitres:

Dans le premier chapitre, on présente des généralités sur les réseaux de capteurs sans fil pour illustrer leurs caractéristiques et leurs spécificités.

Dans le deuxième chapitre, nous parlons des protocoles de routage classiques et ceux basés sur les heuristiques.

Le troisième chapitre, traite des techniques d'optimisation des réseaux en mettant l'accent sur les méta-heuristiques.

Le quatrième chapitre, présente les deux solutions basées sur les algorithmes génétiques qui ont été proposées, la première favorisant le critère énergie et la seconde prenant en considération le critère distance dans l'établissement des chemins. Ensuite, il est question des deux améliorations qui ont été apportés aux algorithmes précédents, La première amélioration porte sur la méthode de sélection alors que la deuxième consiste à ajuster la puissance de transmission au fur et à mesure de la cessation de fonctionnement des nœuds les plus proches de la station de base.

Enfin, on conclut ce mémoire par un récapitulatif sur le travail réalisé dans le cadre de notre projet de fin d'études.

1. Chapitre I Généralités sur les réseaux de capteurs sans fil

1.1. Introduction

Les RCSF sont souvent caractérisés par leur déploiement dense et leurs limitations en termes de ressources telles que la capacité de stockage, de traitement et d'énergie.

Dans ce chapitre, nous donnons une brève présentation des RCSF, leurs caractéristiques et ainsi que les contraintes de routage dans ce type de réseaux.

1.2. Qu'est-ce qu'un capteur ?

1.2.1. Définition d'un nœud capteur

Un capteur est un petit dispositif électronique capable de mesurer une valeur physique environnementale (température, lumière, pression, etc.) ou physiologique (glycémie, tension artérielle, etc.), de transformer cette grandeur en un signal électrique qui peut être interprété par un dispositif de contrôle [1] comme le montre la figure 1.1, et de la communiquer à un centre de contrôle distant via une station de base.

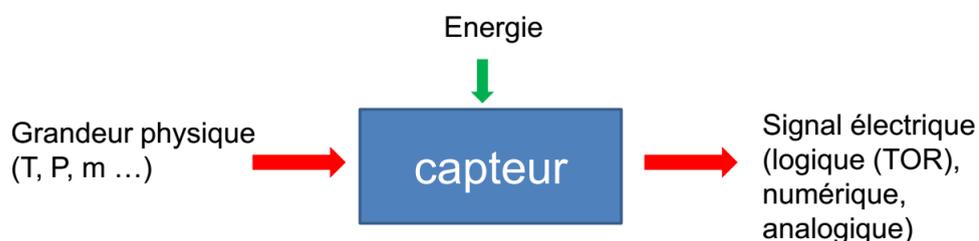


Figure 1.1: Fonctionnement d'un capteur [2]

Un nœud capteur est composé de quatre unités de base [3] (figure 1.2) :

- **L'unité d'acquisition:** est généralement composée de deux sous-unités : les capteurs et les convertisseurs analogique-numériques (ADCs¹). Les capteurs obtiennent des mesures numériques sur les paramètres environnementaux et les transforment en signaux analogiques. Les ADCs convertissent ces signaux analogiques en signaux numériques.

¹ ADC: Analog to Digital Converter ou Convertisseur analogique-numérique.

- **L'unité de traitement:** est composée de deux interfaces: une interface avec l'unité d'acquisition et une autre avec le module de transmission.
- **Un module de communication (Transceiver):** est responsable de toutes les communications via un support de communication radio qui relie un nœud à ses voisins dans un réseau.
- **Batterie:** C'est une source d'énergie par laquelle tous les composants du capteur sont alimentés. Elle correspond généralement à une pile ou une batterie qui s'épuise par le temps. La consommation d'énergie est un facteur critique que doit être pris en compte toutes les applications des capteurs due aux limitations des ressources énergétiques des nœuds. Pour résoudre le problème d'énergie une réalisation récente d'une unité d'alimentation en utilisant des panneaux solaires [4]. Cependant, ce problème persiste puisqu'on n'a pas toujours la possibilité de s'alimenter en énergie solaire. D'où, dans presque tous les travaux de recherche on traite la problématique concernée conjointement avec la consommation d'énergie.

Il existe des capteurs qui sont dotés d'un système de localisation tel que le GPS (Global Positioning System) pour être repéré ou pour repérer les capteurs qui se trouvent dans leur voisinage et d'un mobilisateur leur permet d'être en mouvement comme ceux supporter par les drones.

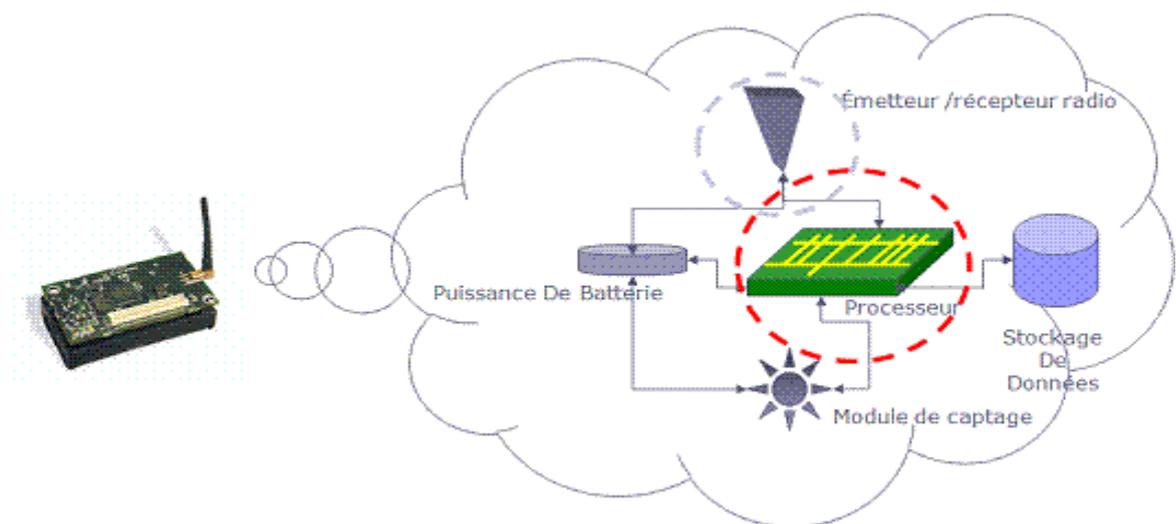


Figure 1.2:structure d'un capteur [5]

1.2.2. Catégories de capteurs

On distingue quatre catégories de capteurs [2] :

- Passif et omnidirectionnel: C'est un capteur qui est doté d'une antenne omnidirectionnel tel que le thermomètre, le microphone, le détecteur chimique, le détecteur de fumée, le détecteur de la pression d'air, ...
- Passif et directionnel: doté d'une antenne de type AntenneRf, photographie.
- Actif : c'est capteur émetteur-récepteur tel que le sonar (instrument de détection sous-marine), le radar, etc...
- Actionneur: C'est un instrument qui réagit suite à l'occurrence d'un événement tel que l'ouverture ou fermeture d'un commutateur, d'un relai pour actionner un moteur, une lampe électrique, ouverture ou fermeture d'une LED² (une diode électroluminescente), etc...

1.3. Les réseaux de capteurs sans fil

Un réseau de capteurs sans fil se compose généralement d'un nombre massives de petits dispositifs appelés "nœuds capteurs ou motes" qui sont dispersés aléatoirement ou d'une manière prédéfinie dans un champ d'intérêt appelé zone de captage en vue de collecter des données et les router directement ou via une communication multi-sauts à un nœud collecteur appelé "station de base ou puits" connecté à un centre de contrôle distant via internet ou bien par satellite, comme le montre la figure 1.3.

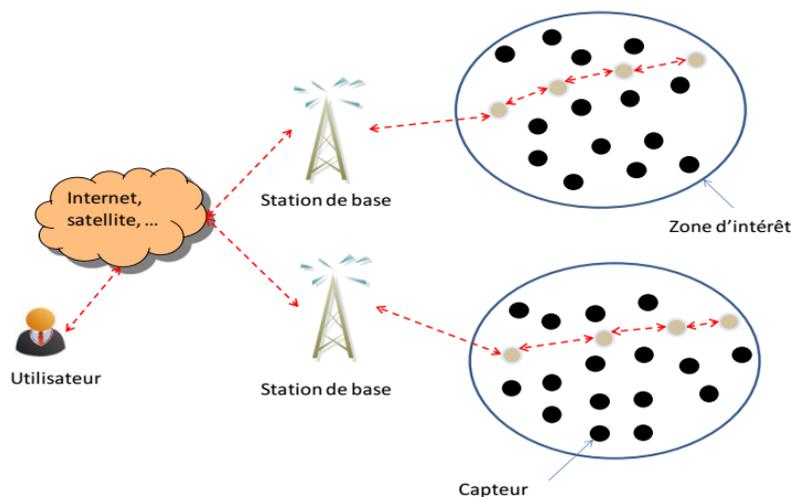


Figure 1.3:exemple de RCSF

² LED: (*Light-Emitting Diode*), est un dispositif opto-électronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique.

Dans un RCSF, les capteurs communiquent entre eux via des liens radio pour le partage d'informations et le traitement coopératif. En outre, l'utilisateur peut adresser ses requêtes aux capteurs en précisant l'information qu'il veut.

1.3.1. Caractéristiques des réseaux de capteurs sans fil

Un RCSF réseau de capteurs présente les caractéristiques suivantes:

- **Absence d'infrastructure** : les réseaux ad-hoc en général, et les réseaux de capteurs en particulier se distinguent des autres réseaux par la propriété d'absence d'infrastructure préexistante et de tout genre d'administration centralisée.
- **Taille importante** : un réseau de capteurs peut contenir un grand nombre de nœuds (d'une dizaine de nœuds à des milliers de nœuds).
- **Interférences** : les liens radio ne sont pas largement espacés dans la bande de fréquences. A cet effet, deux transmissions simultanées sur une même fréquence, ou sur des fréquences proches, peuvent causer des interférences.
- **Topologie dynamique** : les capteurs peuvent être attachés à des objets mobiles qui se déplacent d'une façon libre et arbitraire, et ils sont sujets à des pannes. Ces deux phénomènes rendent ainsi la topologie du réseau fréquemment changeante.
- **Sécurité physique limitée** : les RCSF sont sujets à des attaques plus que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé.
- **Bande passante limitée** : une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un nœud est limitée. A cet effet, la concurrence d'accès au médium pourra avoir une conséquence négative sur d'autres facteurs de performance tel que la latence.
- **Contrainte d'énergie, de stockage et de calcul** : la contrainte la plus critique dans les RCSF est la modestie de ses ressources énergétiques car chaque capteur du réseau possède de faibles ressources en termes d'énergie. Afin de prolonger la durée de vie du réseau, une minimisation de la consommation d'énergie est exigée chez chaque nœud. En outre, la capacité de stockage et la puissance de calcul sont limitées dans un capteur.
- **Portée de transmission** : la portée de transmission des nœuds capteurs dépend de cinq facteurs: la puissance d'émission, la fréquence, la modulation, la localisation et les conditions

météorologiques [6]. Cette portée peut aller de quelques mètres à une centaine de mètres. Par exemple, la portée de transmission de TelosB est de 75m à 100m en outdoor et de 20m à 30m en indoor [7]. De ce fait, les capteurs ne sont pas toujours à la portée radio de la station de base. D'où, ils doivent collaborer entre eux pour envoyer les données collectées jusqu'à la station de base via un schéma de routage multi-sauts. Or, ces schémas de routage appelés encore protocoles de routage peuvent rapidement établir les chemins entre les capteurs et la station de base quand le réseau contient un nombre restreint de capteurs alors qu'ils nécessitent plus de temps quand le nombre de capteurs augmente dans le réseau.

1.3.2. Domaines d'applications

Le faible coût et la communication sans fil permettent aux réseaux de capteurs d'envahir plusieurs domaines d'applications. Ils permettent aussi d'étendre les domaines existants. on peut citer parmi eux:

a. Applications militaires

Le déploiement rapide, l'auto-organisation et la tolérance aux pannes sont des caractéristiques qui ont rendu les réseaux de capteurs efficaces pour les applications militaires.

Un réseau de capteurs peut être déployé dans un endroit stratégique ou hostile, afin de surveiller les mouvements des forces ennemies, ou analyser le terrain avant d'y envoyer des troupes (détection des armes chimiques, biologiques ou radiations et des embuscades).

Le projet DSN³ [8] au DARPA⁴ était l'un des premiers projets dans les années 80 ayant utilisé les réseaux de capteurs pour rassembler des données distribuées. Les chercheurs du laboratoire national Lawrence Livermore ont mis en place le réseau WATS⁵ [9]. Ce réseau est composé de détecteurs des rayons gamma et des neutrons pour détecter et dépister les dispositifs nucléaires. Il est capable d'effectuer la surveillance constante d'une zone d'intérêt. Il utilise des techniques d'agrégation de données pour les rapporter à un centre intelligent. Ces chercheurs ont mis en place ensuite un autre réseau appelé JBREWS⁶ [10] pour avertir les troupes dans le champ de bataille des attaques biologiques possibles. Un réseau de capteurs peut être déployé dans un endroit stratégique ou hostile, afin de surveiller les mouvements des forces ennemies, ou analyser le terrain avant d'y envoyer des troupes (détection des armes chimiques, biologiques ou radiations). L'armée américaine a réalisé des tests dans le désert en Californie.

³ DSN: Distributed Sensor Network

⁴ DARPA: Defense Advanced Research Projects Agency

⁵ WATS: Wide Area Tracking System

⁶ JBREWS: Joint Biological Remote Early Warning System

b. Découvertes de catastrophes naturelles

Par la dispersion des capteurs dans la nature afin de détecter les incendies des forêts, les tempêtes, etc... pour une intervention beaucoup plus rapide.

c. Détection d'intrusions

En plaçant à différents points stratégiques, des capteurs, on peut ainsi prévenir des cambriolages sans avoir à recourir à de coûteux dispositifs de surveillance vidéo.

d. Applications métier

Dans ces applications, le réseau doit pouvoir collecter ces différentes informations et alerter en temps réel si les seuils critiques sont dépassés.

e. Surveillance médicale

En implantant sous la peau des mini capteurs vidéo, on peut recevoir des images en temps réel d'une partie du corps sans aucune chirurgie pendant environ 24h. On peut ainsi surveiller la progression d'une maladie ou la reconstruction d'un muscle.

Les RCSF peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, battements du cœur, ... à l'aide des capteurs ayant chacun une tâche bien particulière. Les données physiologiques collectées par les capteurs peuvent être stockées pendant une longue durée pour le suivi d'un patient [11]. D'autre part, ces réseaux peuvent détecter des comportements anormaux (chute d'un lit, choc, cri, ...) chez les personnes dépendantes (handicapées ou âgées).

f. Contrôle d'édifices

On peut inclure sur les parois des barrages des capteurs qui permettent de calculer en temps réel la pression exercée. Il est donc possible de réguler le niveau d'eau si les limites sont atteintes. On peut aussi imaginer inclure des capteurs entre les sacs de sables formant une digue de fortune. La détection rapide d'infiltration d'eau peut servir à renforcer le barrage en conséquence. Cette technique peut aussi être utilisée pour d'autres constructions tels que ponts, voies de chemins de fer, routes de montagnes, bâtiments et autres ouvrages d'art.

g. Applications environnementales

Le contrôle des paramètres environnementaux par les réseaux de capteurs peut donner naissance à plusieurs applications. Par exemple, le déploiement des thermo-capteurs dans une forêt peut aider à détecter un éventuel début de feu et par suite faciliter la lutte contre les feux de forêt avant leur propagation. Le déploiement des capteurs chimiques dans les milieux urbains

peut aider à détecter la pollution et analyser la qualité d'air. De même leur déploiement dans les sites industriels empêche les risques industriels tels que la fuite de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.).

Le projet HYPOX [12], financé par l'union européenne, a effectué les premiers pas vers l'implémentation d'un système d'observation mondial pour une meilleure compréhension des changements au niveau de l'oxygène dans les systèmes aquatiques. Les chercheurs ont surveillé l'épuisement d'oxygène et les processus associés dans les zones cibles, qui diffèrent au niveau du statut d'oxygène et de sensibilité au changement. Ils ont inclus l'océan arctique, les eaux partiellement enfermées de la mer noire et de la mer baltique, les fjords, les lagons et les lacs enfermés dans les terres. HYPOX a fourni aux décideurs européens les connaissances nécessaires sur l'épuisement d'oxygène dans les systèmes aquatiques. Cela leur permet de développer des stratégies de développement durables efficaces et de négocier des traités internationaux contraignants.

h. La domotique

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, les magnétoscopes, etc. [13] Ces capteurs embarqués peuvent interagir entre eux et avec un réseau externe via Internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance.

1.3.3. Catégories de communications dans les RCSF

La manière d'acheminer les messages dans les réseaux de capteurs peut changer selon l'application qui en est faite. On distingue les catégories de communication suivantes dans les RCSF :

a. Scénario périodique

Dans les scénarios impliquant la collecte de données périodiques, les capteurs transmettent régulièrement leurs mesures à la station de base. Ce processus peut être continu ou pourrait suivre une certaine distribution (géométrique, gaussienne, etc.), déterministe ou probabiliste.

La connaissance du processus d'envoi peut également influencer les périodes de sommeil des capteurs non impliqués dans l'envoi de données. Ce comportement est typique pour effectuer des mesures statistiques d'une certaine métrique, par exemple l'étude du climat.

b. Selon la demande

Lorsque nous souhaitons avoir l'état de la zone de couverture à un moment T, le centre de contrôle émet des broadcasts vers toute la zone pour que les capteurs remontent leur dernier relevé vers le puits. Les informations sont alors acheminées par le biais d'une communication multi-sauts comme s'est illustré par la figure 1.4.

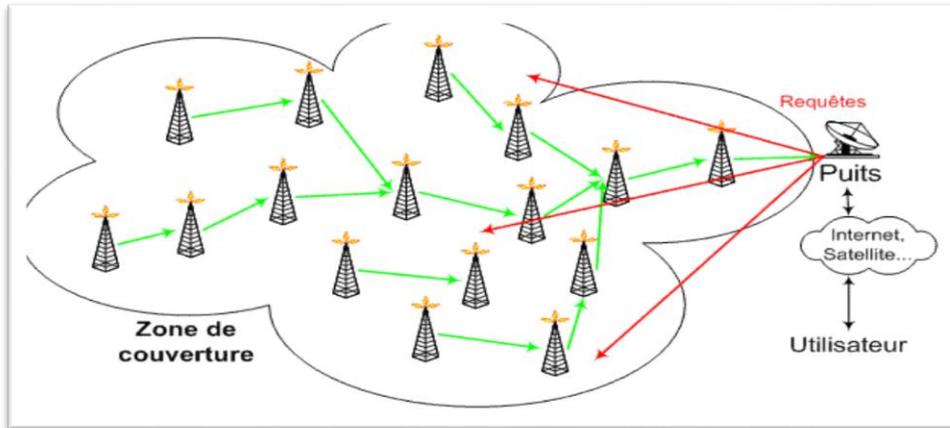


Figure 1.4:collecte d'information à la demande

c. Scénario orienté événement (event-driven)

Dans les scénarios orientés événement (event-driven), les capteurs doivent seulement transmettre une alerte quand un événement pertinent survient ou quand un changement brusque se produit. Dans ce type de scénarios, la réception de message doit être assurée et le délai de transmission doit être limité. Néanmoins, certains mécanismes d'alerte peuvent être désactivés à cause de la défaillance d'un lien radio. Par conséquent, il est recommandé d'envoyer l'alerte plusieurs fois. La figure 1.5 illustre ce type de scénarios.

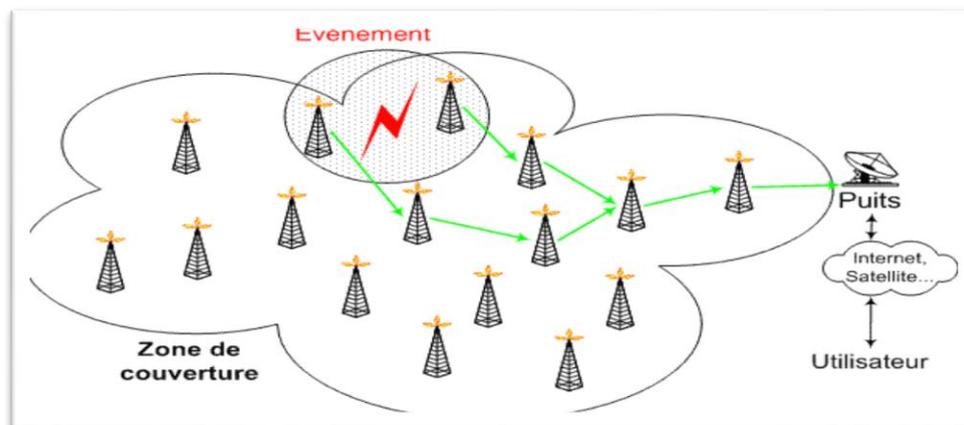


Figure 1.5:Scénario orienté évènement

1.4. Contraintes de routage dans les RCSF

L'acheminement des données est la fonctionnalité la plus importante dans un RCSF. Cette opération doit être supportée par un schéma de routage efficace et fiable. Or, l'adaptation d'une procédure de routage avec un grand nombre de nœuds ayant des capacités de calcul, de sauvegarde et d'énergie limitées; et qui sont déployés dans un environnement caractérisé par des changements de topologies reste un grand défi. A cet effet, toute conception de protocole de routage doit tenir compte des critères suivants :

- **La tolérance de fautes :** Certain nœuds peuvent générer des erreurs ou ne plus fonctionner à cause d'un manque d'énergie, un protocole de routage performant doit éviter l'acheminement des données à travers ces nœuds.
- **Passage à l'échelle :** La plupart des protocoles de routage est conçu à des RCSF de taille moyenne. Or, le nombre de nœuds déployés pour un RCSF peut atteindre des milliers de nœuds. A cet effet, un algorithme de routage performant doit supporter le passage à l'échelle sans voir ses performances se dégradent en cas d'augmentation du nombre des nœuds.
- **L'environnement :** Les capteurs sont souvent déployés en masse dans des endroits tels que les champs de bataille, au-delà des lignes ennemies, à l'intérieur de grandes machines, au fond d'un océan, dans des champs biologiquement ou chimiquement souillés. Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans des régions géographiques éloignées sous des conditions plus ou moins défavorables.
- **La topologie de réseau :** Le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie. Cette maintenance consiste en trois phases : déploiement, post-déploiement (les capteurs peuvent bouger, ne plus fonctionner,...), redéploiement de nœuds additionnels.
- **La consommation d'énergie :** Dans la plupart des cas le remplacement de la batterie d'un capteur est impossible. Ce qui veut dire que la durée de vie d'un capteur décide la durée de vie du réseau. A cet effet, un algorithme de routage fiable doit utiliser les chemins qui minimisent la consommation d'énergie et permet de :
 - Minimiser la charge du réseau en optimisant le nombre d'envois et de réceptions des paquets.
 - Assurer des communications multi-sauts fiables pour éviter les retransmissions.

- Minimiser les envois redondants.
- Auto-organiser le réseau quand un nœud est inséré ou retiré.

1.5. Conclusion

Dans ce chapitre, nous avons présenté les réseaux de capteurs sans fil, leurs caractéristiques et les concepts nécessaires à la compréhension des réseaux de capteurs. En outre, nous avons mis le point sur quelques facteurs permettant la conception des RCSF en particulier l'opération de routage dans les RCSF.

Dans le chapitre qui suit, nous présentons quelques protocoles de routage classique conçus pour les RCSF et d'autres améliorations proposées par l'introduction des méta-heuristiques sur ces derniers.

2. Chapitre II Les protocoles de routages dans les RCSF

2.1. Introduction

Dans les RCSF, on doit assurer la fidélité de détection, de routage et de livraison. La fidélité de détection consiste à assurer la couverture de tout point de la zone d'intérêt par au moins un capteur. La fidélité de routage consiste à établir au moins un chemin entre tout capteur et la station de base alors que la fidélité de livraison permet d'assurer que les messages arrivent aux destinataires sans erreur.

Dans ce chapitre, on présente les critères de performances des protocoles de routage, une classification des protocoles de routage et les principaux protocoles sous-jacents à chaque classe.

2.2. Critères de performance des protocoles de routage dans les RCSF

La spécificité des RCSF a permis d'instaurer des critères de performances bien particuliers pour les protocoles de routage conçus à ce type de réseaux. Parmi ces critères, nous citons [12]:

- **Évolutivité:** l'évolutivité est un facteur important dans les RCSF. Une zone de réseau n'est pas toujours statique, elle change selon les besoins des utilisateurs ou à cause de l'occurrence des pannes. A cet effet, tous les nœuds dans le domaine du réseau doivent être en mesure de s'adapter aux changements de la topologie.
- **L'énergie:** Les capteurs présentent une autonomie d'énergie et les batteries dont ils disposent sont rarement rechargeables et remplaçables. De ce fait, chaque nœud doit économiser la consommation de l'énergie pour assurer ses activités : la détection, le traitement, le stockage et la transmission. Par ailleurs, pour acheminer les paquets vers la station de base, il devra utiliser un protocole de routage performant en termes d'énergie. Ce protocole de routage doit en particulier minimiser le nombre de paquets transmis puisque la transmission est l'opération qui consomme plus d'énergie.
- **Le temps de traitement:** il se réfère au temps pris par le nœud dans le réseau pour assurer l'ensemble des opérations commençant par la détection, le traitement des données, leur

stockage, leur transmission ou leur réception sur le réseau. En outre, l'information devrait y arriver au poste de contrôle dans une durée raisonnable en particulier pour les applications orientées événement.

- **Le schéma de transmission:** la transmission de données par les capteurs vers la destination ou la station de base se fait par un schéma de routage à un seul saut ou à multi-sauts. Par exemple, dans un réseau plat le schéma de routage se fait de nœud en nœud jusqu'à l'arrivée à la station de base. Cependant, dans les protocoles de routage hiérarchiques à l'instar de LEACH les clusterheads communiquent directement l'information à la station de base.
- **Synchronisation :** dans les communications radio entre les nœuds d'un RCSF, les capteurs écoutent en permanence les transmissions et consomment de l'énergie s'ils ne sont pas synchronisés entre eux. Pour cela, un nœud doit savoir gérer son temps de veille en tenant compte des périodes de veille de ses voisins.
- **Overhead:** Avant l'établissement des routes, les capteurs échangent des paquets de contrôle. De ce fait, un protocole de routage performant doit minimiser son overhead. En outre, lors de l'échange de données, on assiste à éviter les collisions pour minimiser les envois multiples.
- **Fiabilité de livraison :** la plupart des protocoles de routage ont été conçus dans un environnement idéal. Néanmoins, la présence des obstacles pour y avoir un impact négatif sur la qualité des messages reçus. De ce fait, il est recommandé de prendre en considération la qualité des liens avant toute communication et le type d'environnement dans lequel ces capteurs sont déployés.

2.3. Taxonomie des protocoles de routage dans RCSFs

En raison de l'évolution du domaine d'applications des RCSF, de nombreux nouveaux algorithmes ont été proposés pour le problème de routage de données dans ce type de réseaux. Ces techniques de routage prennent en considération les caractéristiques des nœuds de capteurs ainsi que les exigences des applications. La majorité de ces protocoles de routage pourra être classée en trois catégories : données centrées (data-centric), hiérarchiques et basés sur la localisation (location-based) comme il existe une quatrième catégorie qui est basée sur les flux de réseau ou la qualité de service (QoS) [15]. Les protocoles de données centrées sont basés sur des requêtes et dépendent de la désignation des données souhaitées, ce qui contribue à éliminer de nombreuses transmissions redondantes. Les protocoles hiérarchiques visent à regrouper les nœuds afin que les clusterheads puissent faire une agrégation des données en vue d'économiser l'énergie. Les protocoles basés sur la localisation utilisent les informations de position pour

transmettre les données vers les régions concernées. La dernière catégorie comprend des approches de routage qui sont basées sur la modélisation des flux et des protocoles ayant des exigences en termes de qualité de service.

Actuellement, on distingue deux types de protocoles de routage : les protocoles de routage classiques qui sont dédiés aux réseaux de capteurs de taille modeste et les protocoles basés sur les heuristiques qui sont conçus aux réseaux de capteurs denses et qui ont dans certains cas des exigences en termes de qualité de service [16]. Ces deux types de protocoles sont classés en quatre catégories qui sont citées avant.

Dans cette section, nous allons explorer les techniques de routage les plus répandues et nous discutons quelques protocoles chacun d'eux dans sa catégorie appropriée. Ceci est dans le but de mieux comprendre les protocoles de routage actuel pour les réseaux de capteurs sans fil et tirer profit de leurs avantages dans notre contribution.

Dans de nombreuses applications de réseaux de capteurs, il est impossible d'attribuer un identifiant à chaque nœud en raison du grand nombre de nœuds déployés. Cette absence d'identification des nœuds ainsi que le déploiement aléatoire des capteurs font qu'il est difficile de sélectionner un ensemble spécifique de capteurs pour être interrogé. Par conséquent, les données sont généralement transmises depuis chaque nœud dans la zone de déploiement avec une redondance importante. Puisque ce processus de transmission de données est inefficace en termes de consommation d'énergie, les protocoles de routage qui seront en mesure de sélectionner un ensemble de capteurs et d'utiliser l'agrégation de données lors de la transmission des données ont été pris en considération. Cette considération a conduit à l'acheminement de données-centrique, qui est différent du traditionnel routage basé sur les adresses où les routes sont créées entre les nœuds adressables.

Dans le routage de données-centrées, la station de base envoie des requêtes à certaines régions et attend les données provenant des capteurs situés dans les régions visées. Puisque les données sont demandées par le biais de requêtes, la précision des attributs est nécessaire pour spécifier les propriétés des données. SPIN [17] est le premier protocole de données-centrées, qui considère la négociation de données entre les nœuds afin d'éliminer les données redondantes et économiser l'énergie. Puis, Directed Diffusion [18] a été développé et est devenu une avancée dans le routage de données-centré. Ensuite, de nombreux autres protocoles ont été proposés soit basés sur le protocole Directed Diffusion [19-21] ou basés sur un concept similaire [22-25].

Nous allons décrire dans ce qui suit, les principaux protocoles faisant partie à cette catégorie en mettant en évidence les idées clés.

2.3.1. Données centrées:

a. SPIN: "Sensor Protocol for Information via Negotiation"

SPIN est l'un des premiers protocoles de routage de catégorie données-centré. SPIN nomme les données en utilisant des descripteurs ou des méta-données. Avant la transmission, les méta-données sont échangées entre les capteurs via un mécanisme d'annonce de données (Data Advertisement), qui est la principale caractéristique de SPIN. Chaque nœud qui reçoit de nouvelles données, annonce à ses voisins cette information et ceux parmi ses voisins qui ne disposent pas de données, envoient un message de demande de données à ce dernier. Ce processus de négociation a permis de résoudre les problèmes classiques causés par le processus d'inondation tels que les données redondantes et le chevauchement des zones de détection. De ce fait, l'efficacité énergétique est améliorée. La figure 2.1 résume le processus de négociation de méta-données dans SPIN. [17]

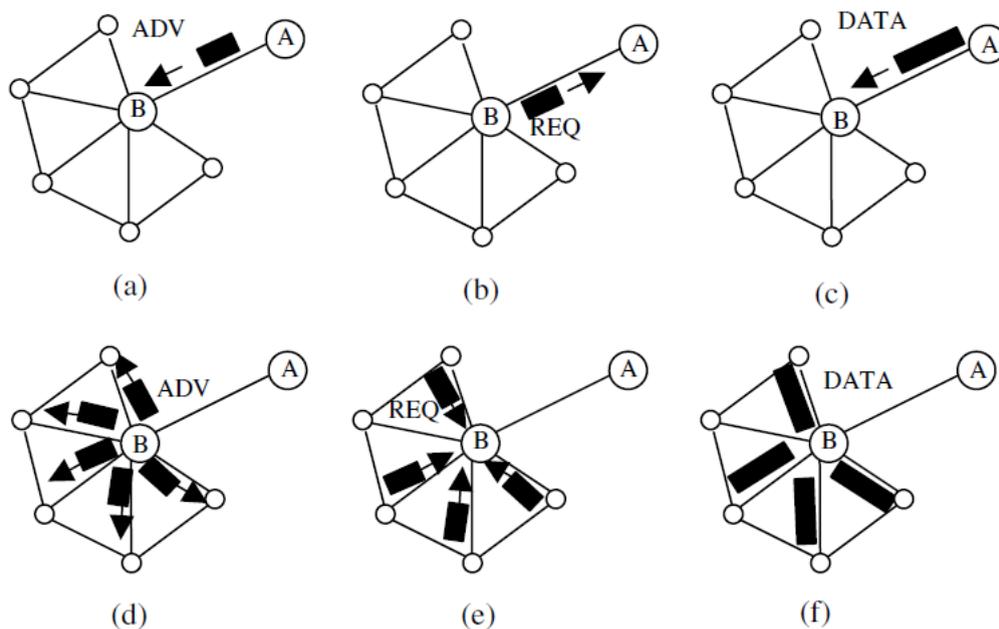


Figure 2.1 : Etapes de fonctionnement du protocole SPIN [17]

b. Directed Diffusion

Diffusion Diffusion [18,26] est un protocole de routage données-centré dédié aux RCFS. DD vise à diffuser des données par le biais de nœuds en utilisant un schéma de nommage pour les données. La principale raison derrière l'utilisation de ce schéma est de se débarrasser des opérations inutiles de la couche réseau afin d'économiser de l'énergie. Diffusion Direct suggère l'utilisation de paires attribut-valeur pour les données et interroge les capteurs en utilisant ces paires. Afin de créer une requête, un intérêt est défini en utilisant une liste de paires attribut-valeur tels que le nom des objets, l'intervalle, la durée, zone géographique, etc. L'intérêt est diffusé par la station de base vers ses voisins et chaque nœud recevant l'intérêt peut faire la mise en cache pour une utilisation ultérieure. Les nœuds ont également la possibilité de faire l'agrégation des données, qui est modélisé par un problème d'arbre minimum de Steiner [27].

Les intérêts dans les caches sont ensuite utilisés pour comparer les données reçues avec les valeurs dans les intérêts. L'entrée d'intérêt contient également plusieurs champs de gradient. Un gradient est un lien de réponse à un voisin dont l'intérêt a été reçu. Il est caractérisé par le débit, la durée et le temps d'expiration provenant des champs de l'intérêt reçu. Par conséquent, en utilisant les intérêts et les gradients, les chemins sont établis entre la station de base et les nœuds sources. Plusieurs chemins peuvent être établis de telle sorte que l'un d'eux est sélectionné par un processus de renforcement positif. La station de base réémet le message d'intérêt initial à travers le chemin d'accès sélectionné à un intervalle plus petit et renforce donc le nœud source sur ce chemin pour envoyer des données plus fréquemment. Figure.2.2 résume le fonctionnement du protocole directed diffusion.

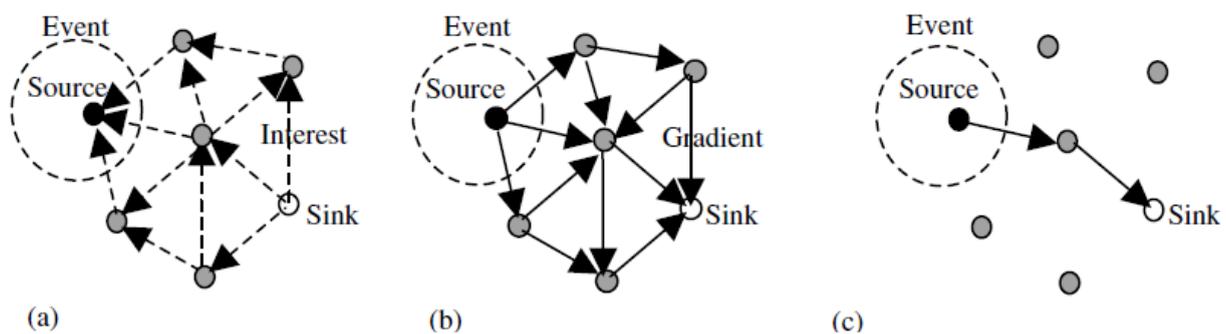


Figure 2.2: Transmission des données dans le protocole Directed diffusion [18]

c. Energy-aware routing

Shah et Rabaey [24] ont proposé d'utiliser un ensemble de chemins semi-optimaux de temps à autre pour augmenter la durée de vie du réseau. Ces chemins sont choisis au moyen d'une fonction de probabilité qui dépend de la consommation d'énergie de chaque chemin. La durée de vie du réseau est la métrique principale que ce protocole la vise.

Le protocole Energy-Aware fait valoir que l'utilisation tout le temps le chemin dont la consommation d'énergie est minimale va épuiser rapidement l'énergie des nœuds sur ce chemin. Pour cela, l'un des chemins multiples est utilisé avec une certaine probabilité, ce qui augmente la durée de vie du réseau. Le protocole suppose que chaque nœud est adressable par un mode d'adressage qui inclut l'emplacement et le type de nœud. Il s'exécute en trois phases:

- **Phase Set-up:** des inondations locales se produisent pour trouver les chemins et créer les tables de routage. En faisant cela, le coût de l'énergie totale est calculé par chaque nœud. Par exemple, si la requête est envoyée à partir du nœud N_i au nœud N_j , N_j calcule le coût de la trajectoire comme suit:

$$C_{N_j, N_i} = Cost(N_i) + Metric(N_j, N_i)$$

La mesure de l'énergie utilisée est calculée en fonction des coûts de transmission et de réception ainsi que l'énergie résiduelle des nœuds. Les chemins qui ont un coût très élevé sont éliminés. La sélection des nœuds est effectuée en fonction de la proximité de la destination. Le nœud attribue une probabilité à chacun de ses voisins dans la table de routage, nommée aussi la table de transfert (FT: Forwarding Table) correspondant aux chemins formés. La probabilité est inversement proportionnelle au coût, à savoir:

$$P_{N_j, N_i} = \frac{1/C_{N_j, N_i}}{\sum_{k \in FT_j} 1/C_{N_j, N_i}}$$

N_j calcule ensuite le coût moyen pour atteindre la destination en utilisant les voisins dans la table de transfert (FTJ) selon la formule suivante:

$$Cost(N_j) = \sum_{i \in FT_j} P_{N_j, N_i} C_{N_j, N_i}$$

Ce coût moyen calculé par N_j est assigné au champ "coût" de la demande et transmis par la suite dans le voisinage de N_j .

- **Phase de communication de données:** Chaque nœud transmet le paquet en choisissant aléatoirement un nœud à partir de sa table de transfert selon certaines probabilités.
- **Phase de maintenance de route:** les inondations locales sont rarement effectuées pour garder tous les chemins opérationnels.

Ce protocole fournit de meilleures performances comparé au protocole Directed Diffusion en termes de consommation d'énergie et de durée de vie. Cependant, ce protocole nécessite la collecte des informations de localisation et la mise en place d'un mécanisme d'adressage pour les nœuds, ce qui complique le passage à l'échelle.

2.3.2. Protocoles hiérarchiques :

Dans une architecture plate, les protocoles perdent leurs performances lors du passage à l'échelle. A cet effet, pour permettre au réseau de faire face à une charge supplémentaire et d'être en mesure de couvrir une grande zone d'intérêt sans dégrader le service, l'architecture hiérarchisée a été adoptée dans certaines techniques de routage.

L'objectif principal du routage hiérarchique est de maintenir efficacement la consommation d'énergie des nœuds en les impliquant dans un schéma de communication multi-sauts et en effectuant l'agrégation de données afin de diminuer le nombre de messages transmis à la station de base. Dans une architecture clustérisée, la formation des clusters est généralement basée sur l'énergie résiduelle des capteurs et de la proximité du capteur de son clusterhead correspondant. [28,29]

LEACH [30] est l'une des premières approches de routage hiérarchique pour les RCFS. L'idée proposée dans LEACH a été une source d'inspiration pour de nombreux protocoles de routage hiérarchiques [22,31-33], bien que certains protocoles ont été développés indépendamment [34,35]. Dans ce qui suit, nous explorons les principaux protocoles de routage hiérarchiques dédiés aux RCFS.

a. LEACH

LEACH est l'un des algorithmes de routage hiérarchiques les plus populaires pour les RCFS. L'idée est de former des clusters basés sur la puissance du signal reçu et utiliser les clusterheads comme des routeurs à la station de base. Ceci permettra d'économiser l'énergie puisque les transmissions ne seront effectuées que par ces clusterheads plutôt que par tous les nœuds.

Tous les traitements de données tels que la fusion ou l'agrégation de données sont assurés par les clusterheads. Ces clusterheads changent de rôle au hasard au fil du temps afin d'équilibrer la dissipation d'énergie des nœuds. Cette décision est prise par le nœud en choisissant un nombre aléatoire entre 0 et 1. Le nœud devient clusterhead pour le cycle actuel si le nombre généré est inférieur au seuil suivant:

$$T(n) = \begin{cases} \frac{p}{1 - p * (r \bmod \frac{1}{p})} & \text{Si } n \in G \\ 0 & \text{Sinon} \end{cases}$$

où p est le pourcentage désiré de clusterheads (par exemple 5%), r est la période courante, et G est l'ensemble des nœuds qui n'ont pas été des clusterheads dans les $1/p$ dernières périodes.

La figure 2.3 illustre la phase de formation de clusters et comment se fait la transmission de données à la station de base dans LEACH.

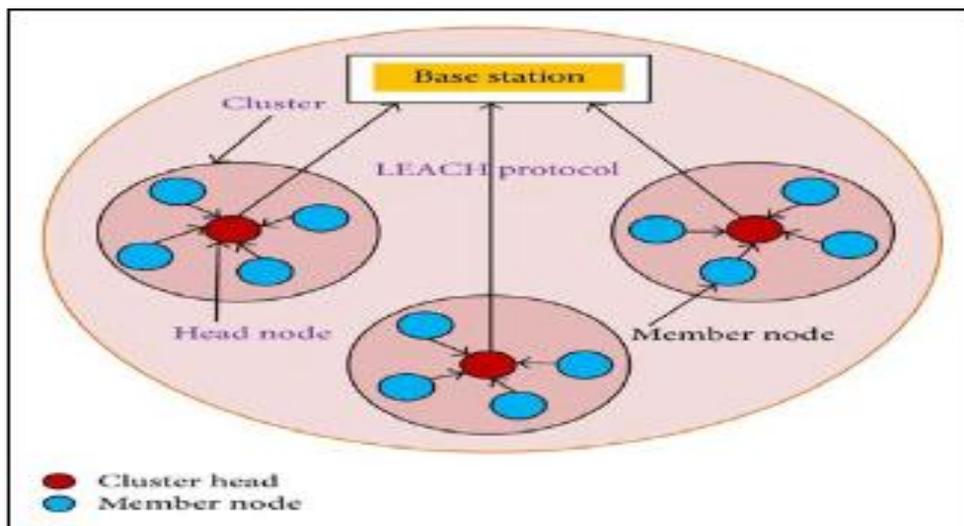


Figure 2.3: Illustration du clustering dans le protocole LEACH [14]

Quoique LEACH fournit de meilleurs résultats, mais l'élection périodique des clusterheads pourra être une charge supplémentaire à cause de l'augmentation des messages d'avertissement lors de l'élection de nouveaux clusterheads, ce qui peut diminuer le gain en consommation d'énergie. [14]

TEEN (Threshold sensitive Energy Efficient sensor Network protocol) [22] a été développé pour modéliser LEACH afin de répondre aux exigences des applications event-driven.

La majorité du comportement de TEEN est semblable au protocole LEACH. Cependant, quelques différences existent entre les deux protocoles par exemple dans le protocole TEEN, les clusterheads élus ne transmettent pas un schedule TDMA aux membres de leurs clusters, mais émettent un message contenant les informations suivantes :

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

Lorsqu'un nœud s'aperçoit que la valeur captée a dépassé le seuil HT, il devra émettre un rapport à son clusterhead correspondant. Il ne réémet un nouveau rapport que si la valeur change radicalement, (i.e: la différence dépasse ST). Ce mécanisme permet d'implémenter un comportement réactif, tout en limitant le nombre de messages utilisés. La figure 2.5 résume le fonctionnement du protocole TEEN.

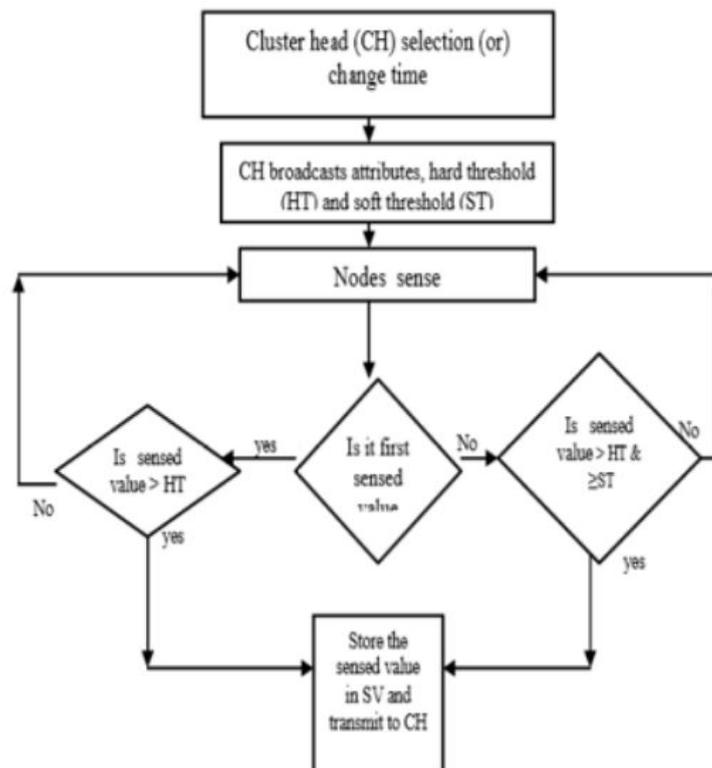


Figure 2.5: Organigramme représentatif du protocole TEEN [22]

d. *CHIRON*

Le protocole CHIRON [36] (Energy-efficient Chain based-Hierarchical Routing Protocol) est classifié parmi les protocoles hiérarchiques. Le routage dans ce protocole se base sur les étapes suivantes:

- **Construction des groupes:** L'objectif principal de cette phase est de diviser le champ de détection en un certain nombre de zones de taille réduite de sorte que CHIRON peut créer plusieurs chaînes plus courtes pour réduire le temps de propagation des données.

Au lieu d'utiliser des clusters concentriques comme dans le protocole EPEGASIS, le protocole CHIRON adopte la technique de BeamStar [37] pour organiser ses clusters. Après que les nœuds de capteurs soient dispersés, la station de base balaie progressivement toute la zone de détection, en changeant successivement des niveaux de puissance de transmission et les directions de son antenne, pour envoyer des informations de contrôle (y compris les valeurs de R et θ) à tous les nœuds. Tous les nœuds recevant les paquets de contrôle, peuvent facilement déterminer le cluster auquel ils peuvent appartenir. En outre, par l'indication d'intensité du signal reçu (RSSI), chaque nœud peut également connaître la distance qui le sépare de la station de base (n_i , BS). Un exemple de clustering avec $R = 1..3$ et $\theta = 1..2$ est représenté par la Figure 2.6.

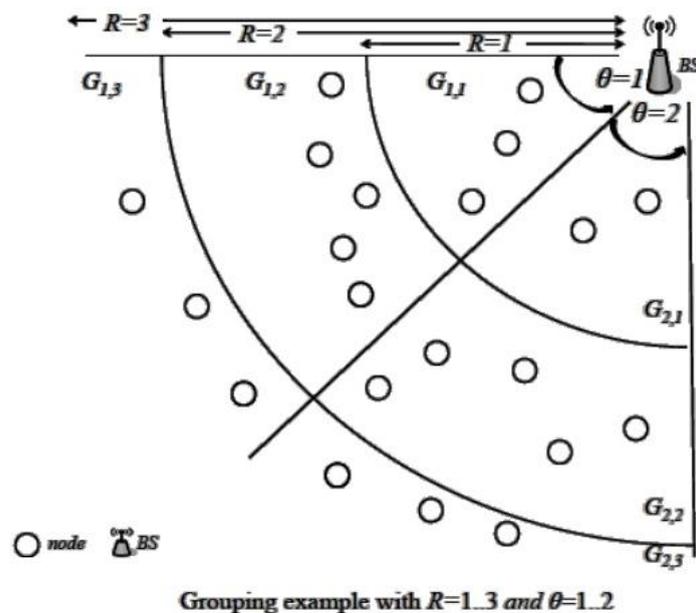


Figure 2.6: Phase de construction des groupes [36]

- **Phase de formation des chaînes :** Dans cette phase, les nœuds de chaque groupe $G_{x,y}$ seront reliés entre eux pour former une chaîne $c_{x,y}$. Le processus de formation de chaînes est le même que celui dans PEGASIS. Pour chaque groupe $G_{x,y}$, le nœud n_i ayant la valeur maximale de $dis(n_i, BS)$ (le nœud est le plus éloigné de la station de base) initie le processus de création de la chaîne du groupe. En utilisant un algorithme glouton, le nœud le plus proche (à n_i) sera choisi pour être lié au nœud n_i , ce dernier initie à son tour la prochaine étape d'établissement de liaison. Le processus est répété jusqu'à ce que tous les nœuds soient mis ensemble, et donc finalement une chaîne du groupe $c_{x,y}$ est formée. La figure 2.7 montre l'ensemble de chaînes de groupe qui sont construits à partir de l'environnement de détection de la figure précédente.

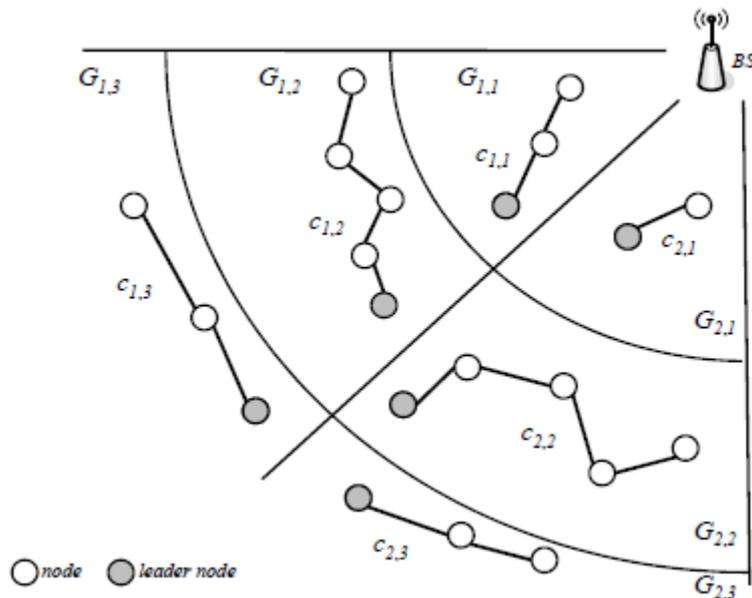


Figure 2.7: Phase d'élection [36]

La phase d'élection d'un nœud chef : Pour la transmission de données, un nœud leader dans chaque chaîne du groupe doit être sélectionné pour la collecte et la transmission des données agrégées à la station de base. Contrairement aux protocoles PEGASIS et EPEGASIS, dans lesquels le chef de dans chaque chaîne est élu d'une manière périodique selon la politique round-robin, CHIRON choisit le leader de la chaîne $(l_{x,y})$ en fonction de la valeur maximale $RES(n_i)$ des nœuds de groupe. Dans un premier temps, dans chaque groupe, le nœud le plus éloigné de la station de base est choisi comme chef de chaîne du groupe. Après cela, pour chaque tour de transmission de données, le nœud ayant l'énergie résiduelle maximale sera élu comme chef de

chaîne du groupe. L'énergie résiduelle de chaque nœud n_i peut être injectée avec les données fusionnées au leader de la chaîne $l_{x,y}$ le long de la chaîne $c_{x,y}$, de sorte que le chef de la chaîne peut déterminer quel nœud sera le nouveau chef de chaîne pour le prochain tour de transmission.

- **Phase de collecte et de transmission de données :** Après les trois phases précédentes, on assiste à la phase de collecte et de transmission des données. La procédure de transmission de données dans CHIRON est similaire à celle dans PEGASIS. Tout d'abord, les nœuds ordinaires dans chaque groupe $G_{x,y}$ transmettent leurs données collectées à leur leader correspondant $l_{x,y}$, en passant par les nœuds les plus proches le long de la chaîne $c_{x,y}$. Et puis, à partir des groupes les plus éloignés, les leaders des chaînes transmettent leurs données agrégées à la station de base selon un schéma de routage multi-sauts (leader-to-leader). Afin d'éviter une distance de transmission plus longue entre deux leaders, et donc donner lieu à une grande quantité de dissipation d'énergie, un leader voisin à $l_{x,y}$ avec les qualifications suivantes sera élu comme nœud relai:

- i) il est plus proche de la base station que $l_{x,y}$;
- ii) la distance à $l_{x,y}$ est minimale. La figure 2.8 montre un exemple d'élection d'un nœud relai et l'acheminement des données par suite, les deux nœuds leaders $l_{1,3}$ et $l_{2,3}$ sélectionnent le leader $l_{2,2}$ comme nœud relais. Tandis que, le nœud leader $l_{2,2}$ envoie ses données fusionnées à son relai $l_{1,2}$. Le processus est répété jusqu'à ce qu'un chemin de transmission en cascade pour atteindre la station de base soit créé.

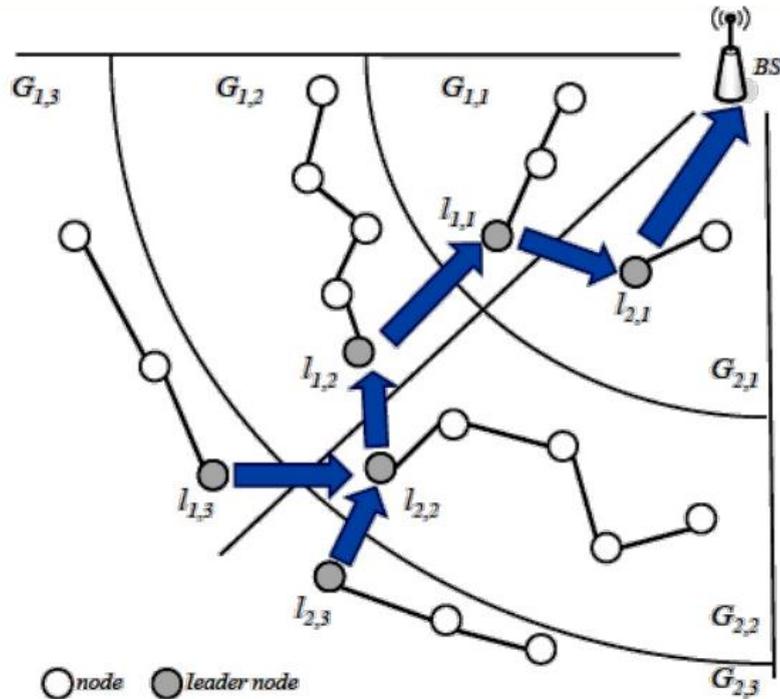


Figure 2.8: Acheminement des données [36]

e. ETR "Energy-aware Tree Routing protocol"

Comme son nom l'indique, ETR [38] utilise un arbre afin d'hiérarchiser le réseau et d'acheminer les données par la suite. ETR s'exécute en trois étapes :

- **Etablissement des routes:** dans cette étape une structure hiérarchique sera créée selon la démarche suivante:
 - La station de base diffuse un message contenant son adresse et son niveau dans la hiérarchie (niveau 0 pour la station de base).
 - Chaque nœud qui reçoit un message, calcule son niveau (niveau=niveau reçu dans le message +1), ajoute de nœud transmetteur dans la liste des parents et de même il diffuse un message contenant son adresse et son niveau.

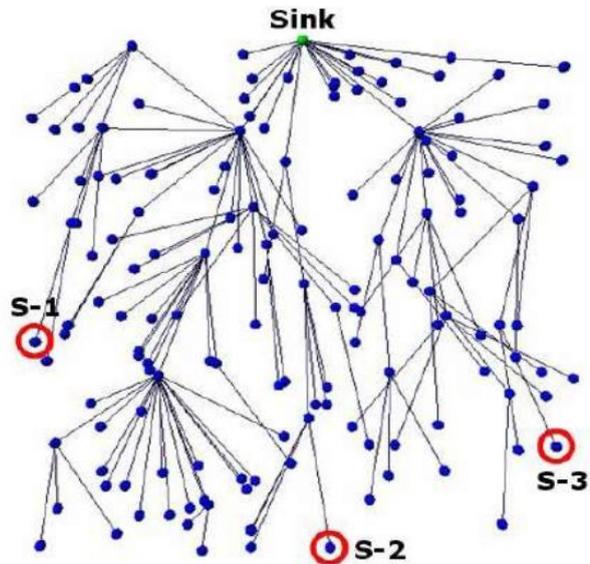


Figure 2.9: Définition des niveaux [38]

- **Transmission des données:** chaque nœud voulant envoyer des données construit un message contenant son adresse et l'envoie à son père, le processus sera répété jusqu'à ce que la donnée arrive à sa destination.
- **Maintenance des routes :** le protocole essaie d'analyser l'énergie des nœuds et reconstruire les liens entre les nœuds et leurs parents si c'est nécessaire.

2.3.3. Les protocoles basés sur la localisation

Dans le routage, certains des protocoles pour les réseaux de capteurs nécessitent des informations de localisation pour les nœuds. Les informations de leur emplacement respectif est nécessaire de manière à faciliter le calcul de la distance entre deux nœuds, et être capable de diffuser une requête sur une région particulière, éliminant ainsi le nombre de transmission [39]. Ceci contribue à l'estimation de la consommation d'énergie.

a. Geographic adaptive fidelity (GAF)

GAF [40] a été conçu principalement pour les réseaux ad-hoc mobiles, mais il est également applicable aux réseaux de capteurs. Il garantit une faible consommation d'énergie dans le réseau en désactivant les nœuds qui ne sont pas impliqués dans le processus de routage sans nécessairement affecter les performances du routage. Chaque nœud est doté d'un GPS lui permettant de connaître son emplacement sur une grille virtuelle.

Dans GAF, les nœuds changent d'état en passant du mode "En dormi" au mode "actif" et vice-versa afin d'équilibrer la charge de consommation d'énergie. Les trois états qui existent dans GAF sont : "découverte" pour connaître l'emplacement des voisins sur la grille, "actif" indiquant la participation des nœuds dans le routage, et "veille" lorsque la radio est éteinte.

GAF peut également être considéré comme un protocole de routage hiérarchique, où les clusters sont basés sur les informations géographiques. Dans chacune des zones de la grille, il existe un nœud principal qui transmet les données vers d'autres nœuds. GAF diffère des protocoles hiérarchiques puisque le nœud leader ne fait pas de l'agrégation ou la fusion des données.

b. Geographic and energy-aware routing (GEAR)

GEAR [41] est un algorithme de routage à haut rendement énergétique conçu pour le routage des requêtes à des régions cibles dans un réseau de capteurs. Dans, GEAR, chaque nœud est doté d'un GPS pour identifier son emplacement. Par ailleurs, GEAR utilise des heuristiques basées sur l'information géographique pour la sélection de nœuds pour acheminer les données vers la station de base. L'objectif principal dans GEAR est de limiter le nombre de cibles dans Directed Diffusion en considérant uniquement une région plutôt que de viser l'ensemble des cibles dans les différentes régions de déploiement des réseaux. En procédant de cette manière, GEAR peut conserver plus d'énergie que le protocole Directed Diffusion.

Dans GEAR, chaque nœud conserve un coût estimé et un coût d'apprentissage pour atteindre la destination à travers ses voisins. Le coût estimé est une combinaison de l'énergie résiduelle et la distance qui le sépare du nœud destinataire. Le coût d'apprentissage est un raffinement du coût estimé qui représente le routage autour des trous dans le réseau. Un trou se produit quand un nœud n'a pas de voisin proche dans la région cible. S'il n'y a pas de trous, le coût estimé est égal au coût apprentissage.

Le coût d'apprentissage se propage un saut en arrière chaque fois qu'un paquet atteint la destination afin que la configuration de la route pour le prochain paquet sera ajusté. GAER s'exécute en deux phases:

- **Transfert des paquets vers la région cible:** à la réception d'un paquet, un nœud vérifie ses voisins pour voir s'il y a un voisin qui est plus proche de la région cible que lui-même. S'il y a plusieurs qui sont proches, le plus proche voisin de la région cible est choisi comme saut suivant. Si tous ses voisins sont plus loin de la région cible que le nœud lui-même, cela signifie qu'il y a un trou. Dans ce cas, l'un des voisins est prélevé pour transmettre le paquet

sur la base de la fonction de coût d'apprentissage. Ce choix peut alors être mis à jour en fonction de la convergence du coût d'apprentissage lors de la livraison de paquets.

- **La transmission des paquets au sein de la région:** si le paquet a atteint la région, il peut être diffusé dans cette région géographique soit par le transfert récursif ou l'inondation restreinte. L'inondation restreinte est bonne lorsque les capteurs ne sont pas densément déployés. Dans les réseaux à haute densité, la transmission géographique récursive est économe en énergie que l'inondation restreinte. Dans ce cas, la région est divisée en quatre sous-régions et quatre copies du paquet sont créées. Ce processus de fractionnement et transfert continue jusqu'à ce que les régions avec un seul nœud soient épuisées. GEAR ne réduit pas seulement la consommation d'énergie, mais surpasse également GPSR [42] en termes de livraison de paquets.

2.3.4. Principaux protocoles de routage basés sur les heuristiques

Les protocoles de routage classiques ont prouvé leur faiblesse lors du passage à l'échelle dans les réseaux de capteurs en termes de latence, durée de vie, etc. A cet effet, de nouvelles approches basées sur les heuristiques ont été révélées pour surpasser les failles présentées par les algorithmes de routage classiques. Parmi ces algorithmes, nous trouvons la colonie de fourmis, le recuit simulé, les algorithmes génétiques, la méthode tabou. Dans ces heuristiques, il y avait l'imitation du comportement des espèces naturelles telles que les fourmis.

Les colonies de fourmi sont l'un des heuristiques les plus utilisées dans la procédure de routage. Le routage basé sur ces dernières représente une approche prometteuse, où le comportement des fourmis et la communication entre elles en se basant sur l'adoption de produits chimiques comme substance connue sous le nom de phéromones paré très intéressante et peut être projeté dans les réseaux à grandes échelles [43].

a. CRP "*Comprehensive Routing Protocol* "

Dans l'objectif d'améliorer le protocole EAR, le protocole CRP [44] a été fondé en se basant sur l'utilisation de l'heuristique colonie de fourmi.

Ce protocole repose sur trois phases importantes :

- **Configuration de la table de routage :** dans cette phase une exploration de tous les chemins qui relient la source à une destination sera déclenchée. Ensuite et pour chaque chemin trouvé le protocole lui attribue une certaine probabilité selon les critères suivants:

- la quantité des phéromones.
 - l'énergie restante de chaque un des nœuds qui forment le chemin.
 - la fréquence qu'un nœud sera considéré comme étant un routeur.
- **Communication de données** : le nœud source envoie le paquet de données à l'un de ses voisins de la table de transfert, avec la probabilité choisie par le voisin. Chacun des nœuds intermédiaires transmet le paquet de données vers un voisin choisi aléatoirement dans sa table de transfert, également avec la probabilité du voisin étant choisie égale à la probabilité de la table de transfert. Ceci se poursuit jusqu'à ce que le paquet de données atteigne le nœud de destination. Durant la transmission de données, quand un nœud est choisi comme un nœud relai, la concentration de phéromone sur la branche reliant ces deux nœuds doit être mise à jour selon les équations suivantes:

$$l_{ij} = l_{ij} + \Delta l_{ij}$$

Où l_{ij} est la quantité de phéromone sur chemin reliant i et j et Δl_{ij} est la quantité de phéromone de mise à jour.

- **Maintenance de routes** : Cette phase est responsable de rétablir les routes et mettre à jour des tables de routage. Des inondations locales sont effectuées fréquemment de la destination à la source pour maintenir tous les chemins opérationnels et mettre à jour les tables de routage en fonction des conditions actuelles.

Les résultats fournis par CRP ont montré qu'il dépasse ceux du protocole EAR. Cependant, CRP manque des métriques de qualité de service.

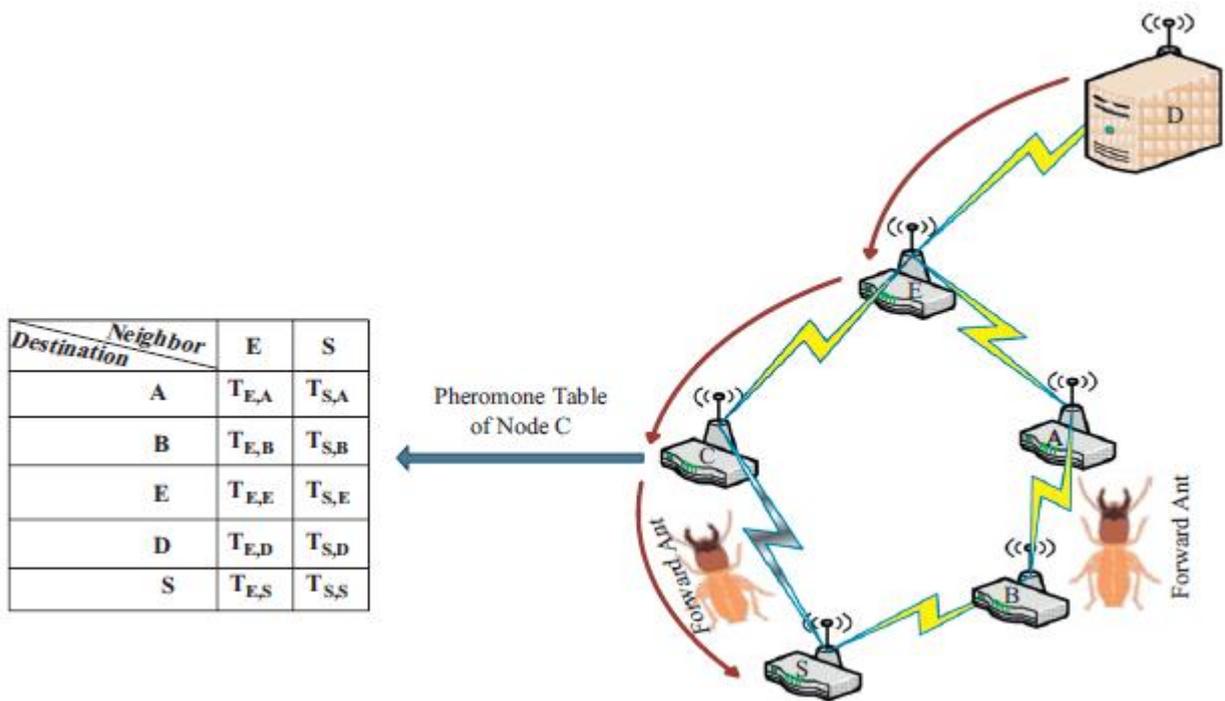


Figure 2.10:Exemple d'une table de routage du protocole CRP [44]

b. ACALEACH: " Ant Colony clustering algorithm "

Wang et al. [45] ont proposé un algorithme clustérisé basé sur la colonie de fourmis, appelé ACALEACH. Ce dernier est une version améliorée de LEACH.

ACALEACH ne considère pas seulement l'énergie résiduelle des nœuds lors de l'élection des CHs, mais aussi la distance entre les clusterheads. En outre, il applique l'approche de la colonie de fourmis dans le schéma de routage inter-cluster pour réduire la consommation d'énergie des clusterheads et par conséquent prolonger la durée de vie des réseaux de capteurs.

Les résultats de simulation ont montré que le protocole ACALEACH est meilleur que LEACH en termes de consommations d'énergie et du nombre de nœuds vivants à la fin de la simulation. Cependant, ACALEACH n'a pas évalué le débit et la latence dans son processus de routage, et il peut être faible dans l'efficacité énergétique en raison de l'overhead lors du passage à l'échelle.

c. MSRP: " Multi-SinkRoutingprotocol "

MSRP [46] est un protocole de routage pour les réseaux de capteurs qui est auto-organisé, tolérant aux pannes et adaptable à plusieurs types d'environnement. Le protocole est inspiré du comportement des petits organismes. Ces organismes trouvent leur avantage dans la capacité de

s'organiser en clusters en utilisant la génération de phéromone et son évaporation. Le protocole organise le trafic de données vers la station de base en adoptant le concept de gradient tout en montrant l'autonomie et la tolérance aux pannes.

L'évaluation de MSRP donne de meilleures performances en termes d'overhead et l'adaptation aux changements dans l'environnement. La figure 2.11 illustre les étapes du procédé de signalisation de l'algorithme.

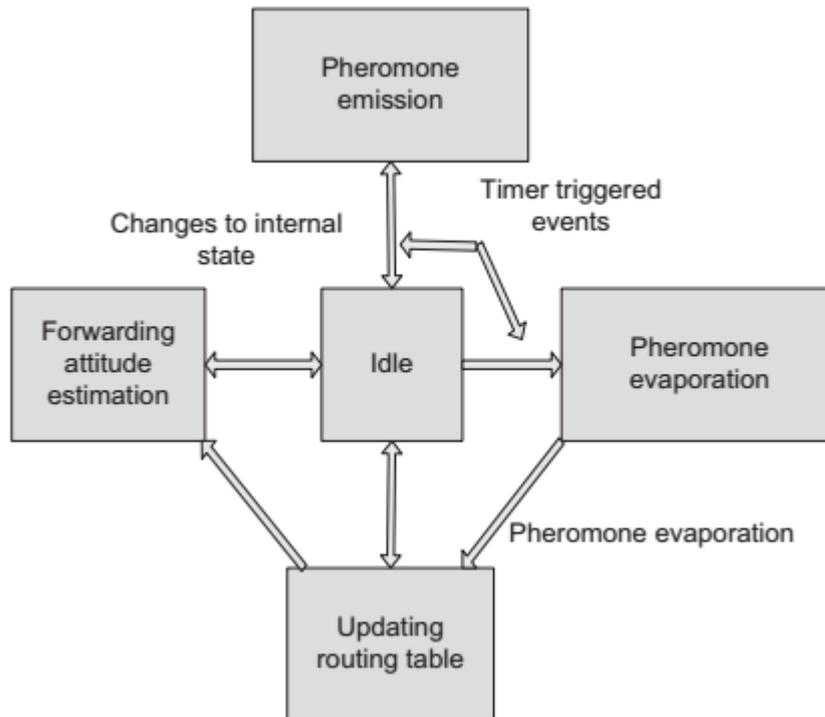


Figure 2.11: Illustration du fonctionnement du protocole MSRP [46]

d. Ant colony multicast trees (ACMT)

De-min et al. [47] ont proposé le protocole ACMT qui est basé les arbres multicast en utilisant la colonie des fourmis pour le routage dans les RCSF. L'algorithme essaie de prolonger la durée de vie du réseau en minimisant la consommation d'énergie dans le processus de communication. Dans ACMT, les fourmis ont trouvé arbres, dont l'arbre que la fourmi a trouvé comprend des nœuds de destination. Il n'y a pas un seul nœud courant pour chaque fourmi. Chaque nœud de l'arbre qui a été trouvé est susceptible d'être le nœud courant.

Dans ACMT, le seul principe observé par le nouvel algorithme est le mécanisme de rétroaction positive de l'algorithme de la colonie de fourmis. Les simulations ont montré que le protocole ACMT fonctionne mieux comparé au modèle YANG et à l'inondation. Cependant,

comme le réseau se développe plus rapidement, le taux de nœuds qui cessent de fonctionner devient plus élevé.

2.4. Conclusion

Dans ce chapitre, nous avons présenté une panoplie de protocoles de routage pour les RCSF et nous avons classifié ces protocoles en deux catégories : les protocoles de routage classiques et les protocoles de routage basés sur les heuristiques.

Cette exploration nous a permis de comprendre comment fonctionnent les principaux protocoles de chaque catégorie et nous a permis également de tirer profits de leurs avantages et d'éviter leurs inconvénients. Par ailleurs, nous avons remarqué que les protocoles de routage classiques perdent leurs performances quand le nombre de nœuds augmente dans le réseau et pour engendrer cette limitation les heuristiques sont considérées comme une approche prometteuse.

Dans le chapitre suivant, nous allons présenter les techniques utilisées pour résoudre les problèmes NP en mettant l'accent sur les heuristiques.

3. Chapitre III Techniques d'optimisations pour les réseaux : les heuristiques

3.1. Introduction

La plupart des protocoles de routage conçus pour les RCSF ont été dédiés pour des réseaux de taille moyenne. Cependant, lors du passage à l'échelle ces protocoles perdent leurs performances. De ce fait, il s'avère nécessaire de développer ces protocoles dans un contexte qui leur permet de garder plus au moins leurs performances sans les voir dégrader d'une manière chaotique.

Pour les problèmes de routage à grande échelle dans les RCSF, il n'existe aucun algorithme connu ayant un temps d'exécution polynomial. Les seuls algorithmes qui permettent d'obtenir une solution optimale ont une complexité asymptotique très élevée (complexité exponentielle $\theta(2^n)$ ou factorielle $\theta(n!)$).

Par ailleurs, la recherche d'une solution exacte optimale à l'aide des algorithmes de complexité exponentielle ou factorielle est donc impensable sauf si on dispose de beaucoup de ressources de calcul. Par exemple, en 1997, le record pour le PVC pour une instance comptant 7397 villes, avait demandé 3 ans de temps CPU sur un réseau de stations SUN. En 2004, 24 978 villes ça demande le même temps.

La théorie des problèmes NP-difficiles porte sur une classe de problèmes pour lesquels il n'existe aucun algorithme connu qui a une complexité polynomiale. Pour ces problèmes difficiles, puisqu'il est impossible d'obtenir une solution dans un temps raisonnable, on fait recours à des algorithmes qui fournissent des solutions "pas trop mauvaises".

Dans ce chapitre, nous présentons les classes d'algorithmes pour les problèmes NP_difficiles tout en mettant le point sur les heuristiques.

3.2. Classes d'algorithmes pour les problèmes NP_difficiles

On distingue quatre classes:

3.2.1. Les algorithmes d'approximation

Les algorithmes d'approximation permettent d'obtenir une solution pas trop éloignée de la solution optimale.

On réserve le qualificatif "algorithme d'approximation" aux algorithmes pour lesquels on peut définir de façon précise la notion "pas de trop éloignée de la solution optimale".

3.2.2. Les algorithmiques heuristiques

Les algorithmes heuristiques produisent une solution qui approxime la solution optimale. Dans ces algorithmes, il n'est pas nécessairement possible de borner de façon exacte la qualité de la solution produite relativement à la solution optimale. En outre, une heuristique pourra produire une solution qui n'est pas bonne ou ne produire aucune solution.

Les heuristiques peuvent être classées en deux catégories:

- Les méthodes constructives: ces méthodes génèrent des solutions de façon incrémentale. Donc, à partir d'une solution initialement vide à laquelle sont ajoutés des éléments jusqu'à l'obtention d'une solution plus ou moins complète.
- Les méthodes de fouilles locales: ces méthodes partent d'une solution initialement complète et de façon répétitive tente d'améliorer cette solution en explorant son voisinage immédiat.

3.2.3. Les algorithmes probabilistes

Dans ces algorithmes, quand on est confronté à un choix, ce choix se fait aléatoirement. En outre ces algorithmes ont un caractère non déterministe.

3.2.4. Les méta-heuristiques

Une stratégie de résolution de problème qui utilise, qui coordonne d'autres heuristiques pour obtenir une solution à un problème difficile.

Une heuristique est conçue pour un problème précis alors que les méta-heuristiques sont conçues pour s'appliquer à divers problèmes.

3.3. Exemple de méta-heuristique

Dans ce projet de fin d'études nous nous intéressons aux méta-heuristiques en particulier, pour éclaircir de plus en plus le fonctionnement de ces techniques d'optimisations nous citons les approches suivantes:

3.3.1. La colonie de fourmis

Présentation :

Cette méta-heuristique a été introduite en 1991 pour résoudre le problème du voyageur de commerce. Elle a été l'objet d'améliorations dès 1995 et a été appliquée avec succès à d'autres problèmes d'optimisation combinatoire [48].

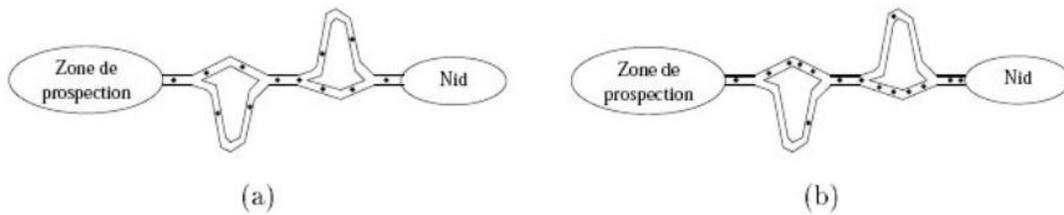


Figure 3.1: Expérience du double pont binaire [48]

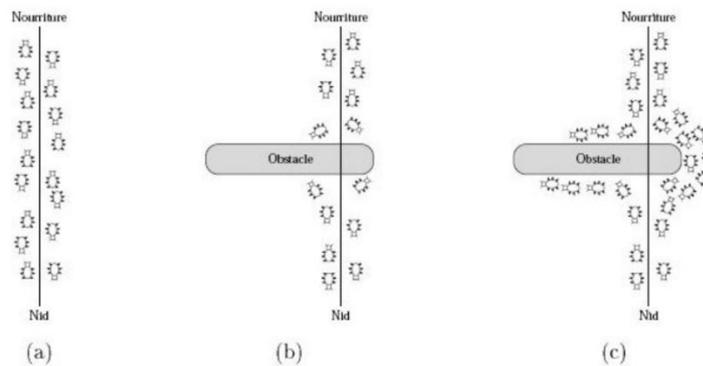


Figure 3.2: Effet de coupure d'une piste de phéromones [48]

Caractéristiques des fourmis virtuelles :

Dans cette sous-section, nous présentons une analogie entre les fourmis réelles et les fourmis virtuelles [49]:

- Comme pour les fourmis réelles, une colonie virtuelle est un ensemble d'entités non-synchronisées, qui se rassemblent pour trouver une bonne solution au problème traité. Chaque groupe d'individus doit pouvoir trouver une solution même si elle est mauvaise.
- Ces entités communiquent par le mécanisme des pistes de phéromone. Cette forme de communication joue un grand rôle dans le comportement des fourmis et de changer la manière dont l'environnement est perçu par ces dernières.

- La méta-heuristique offre la possibilité d'évaporation des phéromones. Ce mécanisme permet d'oublier lentement ce qui s'est passé avant. C'est ainsi qu'elle peut diriger la recherche vers de nouvelles directions.
- les fourmis doivent décider leur prochain déplacement. Cette prise de décision se fait au hasard et dépend de l'information locale déposée sur le site courant.

Implémentation générale d'un algorithme de colonie de fourmis :

La méta-heuristique de colonie de fourmis s'exécute selon le schéma algorithmique suivant:

```

Initialiser les pistes de phéromones ; ;
while critère de fin non rencontré do
  | for chaque fourmi do
  | | Construction de la solution en utilisant les piste de phéromones
  | | Mettre à jour les pistes :
  | | - Evaporation ;
  | | - Renforcement ;
  | end
end
Sortie : meilleure solution trouvée ou un ensemble de solutions.
    
```

Figure 3.3:Pseudocode de l'heuristique colonie de fourmi [49]

En plus de la construction d'un chemin valide, la mise à jour des phéromones est l'une des étapes les plus importantes de l'algorithme où elle décide le comportement future de chaque fourmi. Cette étape est effectuée selon une règle globale et constitué de deux phases essentielles:

- **Phase d'évaporation:** la quantité des phéromones présente dans chaque chemin sera réduite d'un pourcentage fixe selon la formulation suivante :

$$\tau_{ij} = (1 - \rho)\tau_{ij}, \quad \forall i, j \in [1, n] \quad \text{et } \rho \in]0,1[$$

Où ρ représente le taux de réduction de la phéromone

- **Phase de renforcement:** la piste des phéromones sera mise à jour en fonction des solutions explorées. Dans cette étape on distingue trois différentes façons de mise à jour [49]:

- **Mise à jour en ligne:** la quantité des phéromones est modifiée au fur et à mesure de la construction d'une solution.

- **Mise à jour en ligne retardée:** la quantité des phéromones sera modifiée lors de la construction d'une solution complète.
- **Mise à jour hors ligne:** la quantité des phéromones sera modifiée après la construction de toutes les solutions. Cette manière est la plus utilisée car elle offre plusieurs possibilités lors du changement des pistes de phéromones où on peut favoriser les meilleures solutions obtenues, pénaliser les mauvaises ou bien effectuer une mise à jour élitiste.

$$\tau_{i\pi^*(i)} = \tau_{i\pi^*(i)} + \Delta, \quad \forall i, j \in [1, n]$$

Les fourmis élitistes :

Les fourmis élitistes est une amélioration de l'algorithme de colonie de fourmis, l'idée est de renforcer la quantité des phéromones dans chaque arc appartenant à la meilleure solution trouvée après chaque itération.

Cette méthode consiste à faire parcourir le meilleur tour par certaines fourmis artificielles, dites élitistes. Ces fourmis sont conceptuellement identiques aux autres fourmis, la seule particularité c'est qu'elles choisissent leur chemin d'une manière déterministe et réalise l'opération suivante [49]:

$$\tau_{ij}(t) \leftarrow \tau_{ij}(t) + \begin{cases} e \cdot \frac{Q}{L^*} & \text{si } (i, j) \in T^* \\ 0 & \text{sinon} \end{cases}$$

e est le nombre de fourmis élitistes.

3.3.2. La méthode Tabou

La recherche Tabou est une méta-heuristique d'optimisation présentée par Fred W.Glover en 1986 et introduit une notion de mémoire dans la stratégie d'exploration de l'espace de recherche [50].

Principe :

L'idée de la recherche Tabou consiste à explorer le voisinage et à choisir le meilleur voisin trouvé et de refaire la même procédure jusqu'à ce qu'on arrive à un état de stagnation.

La sélection du meilleur voisin s'effectue soit en explorant le voisinage en entier et dans ce cas on parle de la méthode "Best fit", soit le meilleur parmi une partie du voisinage et dans ce cas on parle de la méthode "First fit" [50,51].

La figure 3.4 montre comment se fait l'exploration d'un voisinage pour trouver le meilleur voisin.

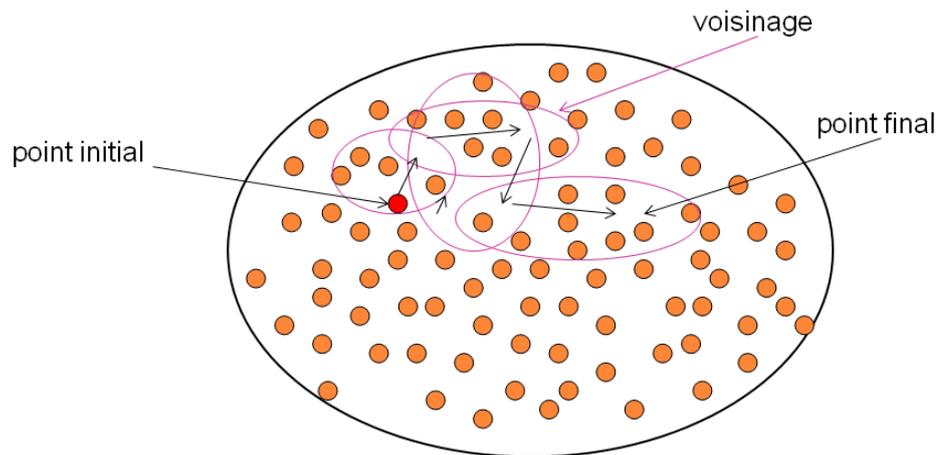


Figure 3.4: Illustration d'une recherche locale [51]

Cependant cette méthode de recherche peut causer ce qu'on appelle le problème de l'optimum local, l'heuristique Tabou corrige ce problème avec deux mécanismes :

- accepter la dégradation selon un seuil prédéfini.
- mettre les solutions explorées dans une liste ("la liste tabou") et interdire leurs utilisations durant un nombre prédéfinis d'itérations [51].

Notion de mémoire de recherche :

Cette notion consiste à mémoriser les solutions explorées et les utilisées par suite pour améliorer la recherche, on distingue trois types de mémoires :

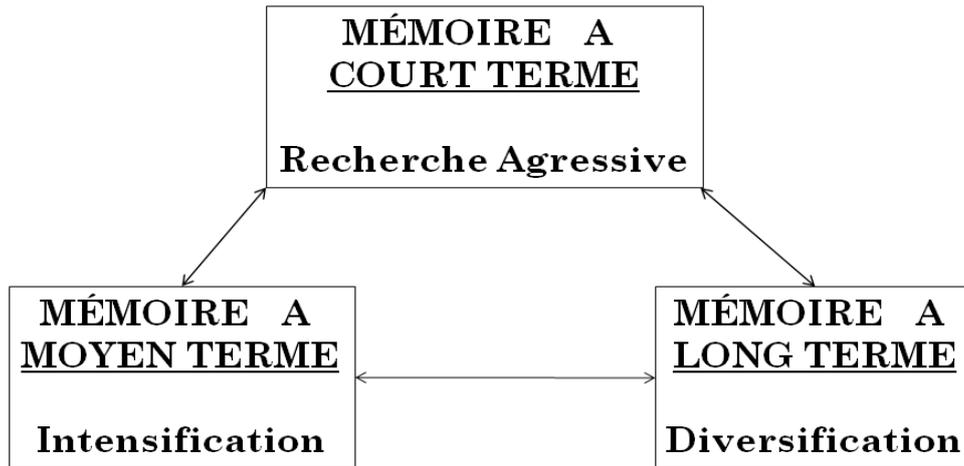


Figure 3.5: Types de mémoires de recherches [51]

- **Mémoire à court terme:** basée sur le critère d'aspiration qui permet de lever le statut tabou sur une solution si elle améliore la meilleure solution obtenue durant la recherche.
- **Mémoire à moyen terme "Intensification":** consiste à mémoriser les meilleures solutions explorées, tirer les points communs entre elles afin de diriger la recherche vers des zones prometteuses.
- **Mémoire à long terme "Diversification":** l'idée est de mémoriser les solutions les plus visitées et leurs imposer un système de pénalité.

La figure 3.6 illustre le comportement des mémoires de recherche.

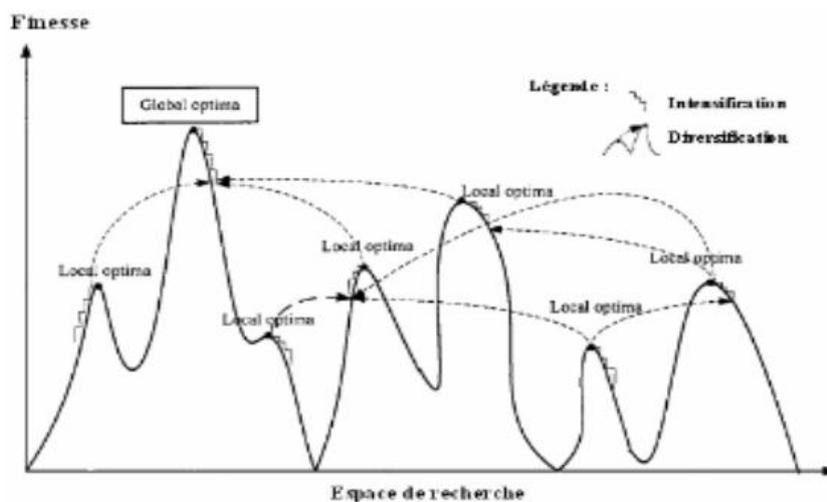


Figure 3.6: Comportement des mémoires de recherche [50]

Implémentation de la recherche Tabou :

L'implémentation de l'algorithme de recherche tabou suit l'organigramme suivant :

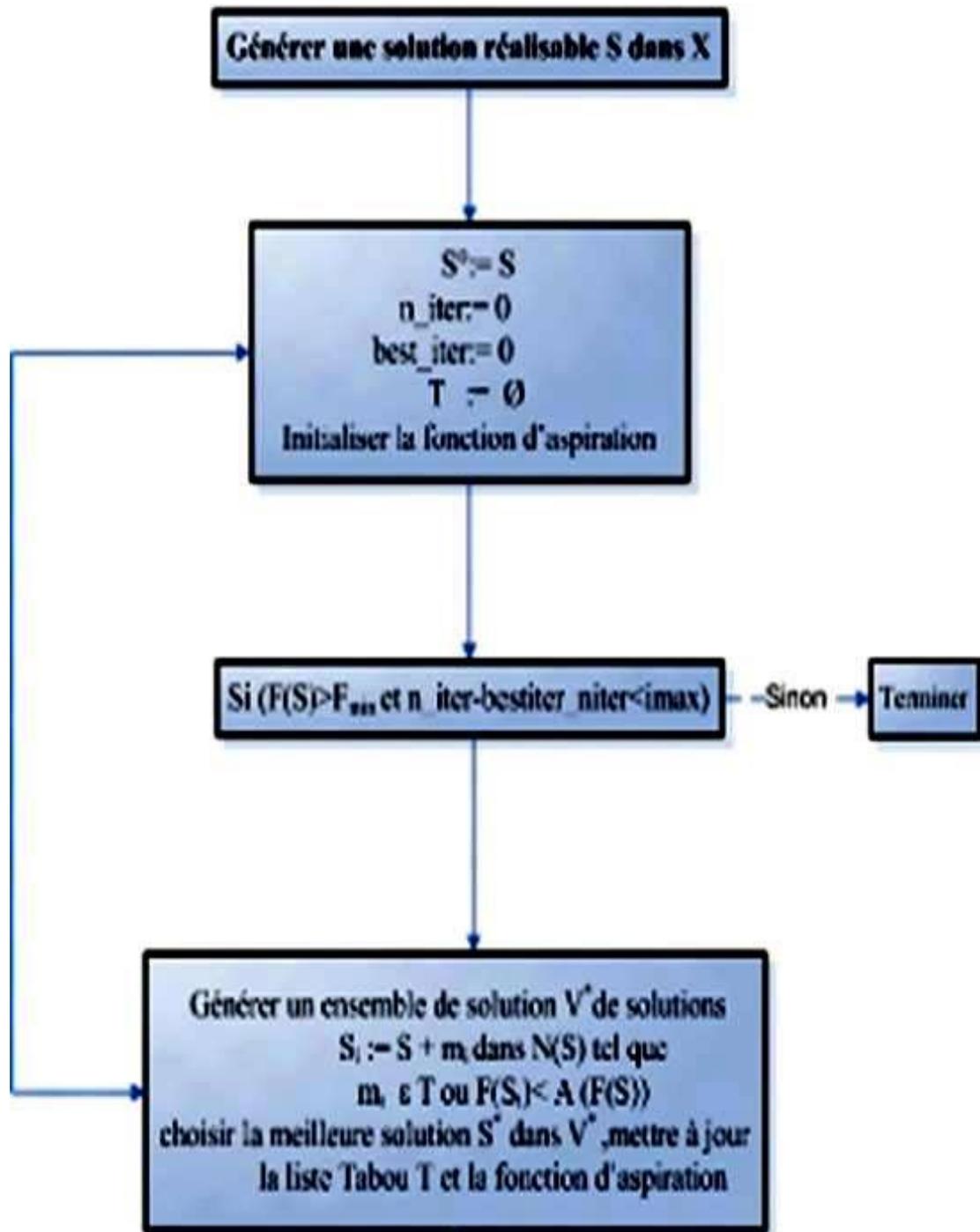


Figure 3.7: Organigramme d'une recherche Tabou

3.3.3. Le recuit simulé

Cette méta-heuristique a été mise au point par trois chercheurs de la société IBM, S. Kirkpatrick, C.D. Gelatt et M.P. Vecchi en 1983, et indépendamment par V. Cerny en 1985.

Le recuit simulé est une méthode empirique inspirée de l'événement de refroidissement naturel de certains métaux où l'état stable sera abouti par une maîtrise du processus de refroidissement [50].

Principe :

Pour chaque température l'heuristique explore un ensemble de solutions voisines, dans ce cas on sera face à deux situations:

- la solution explorée est meilleure que la solution actuelle par conséquent elle sera choisie automatiquement.
- la solution explorée représente une dégradation de la solution actuelle mais elle pourra être choisie avec une certaine probabilité calculée en fonction de la valeur de la dégradation de la solution et la température.

La figure 3.8 illustre les étapes du déroulement du recuit simulé.

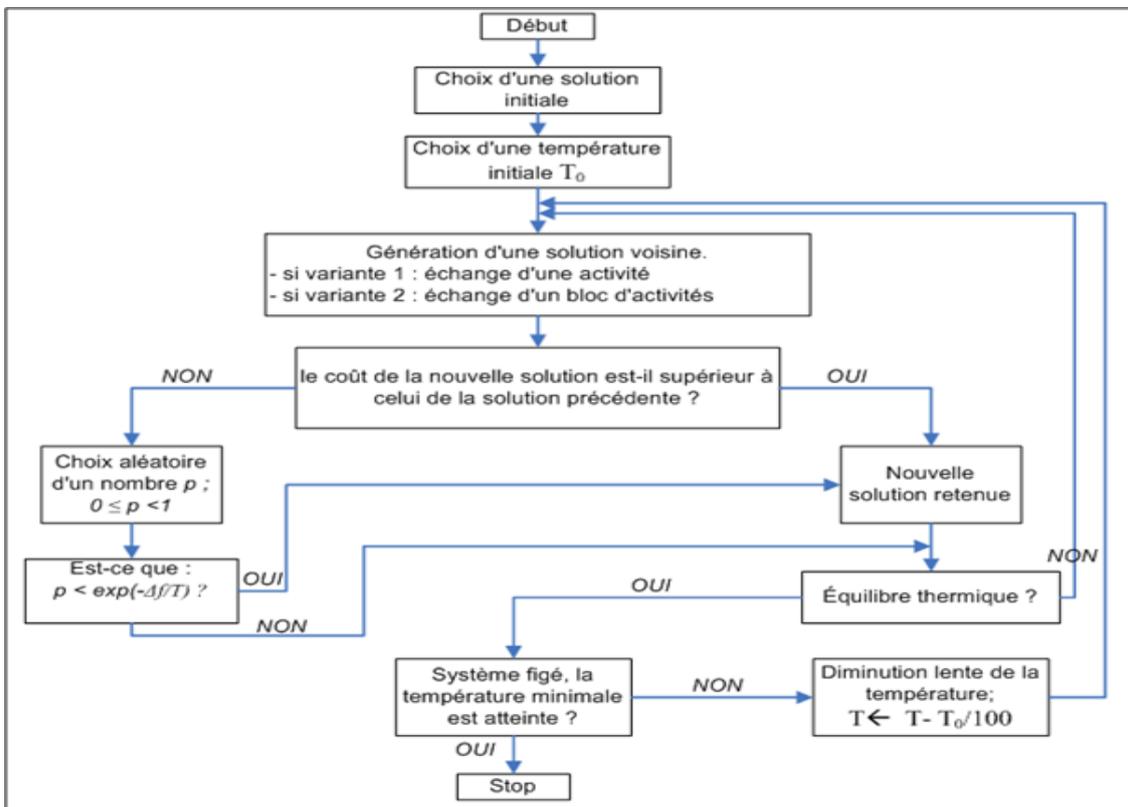


Figure 3.8: Organigramme représentatif du recuit simulé

3.3.4. Les algorithmes génétiques

L'utilisation des algorithmes génétiques est à l'origine le fruit des recherches de John Holland et de ses collègues et élèves de l'université de Michigan. Ils sont inspirés des phénomènes biologiques et de la théorie d'évolution en particulier. Ce type d'algorithmes a été conçu afin de résoudre des problèmes jugés NP_difficiles [52].

Principe :

Le principe général des algorithmes génétiques consiste à améliorer une population initiale en se basant sur plusieurs méthodes pour modéliser, évaluer la qualité des solutions obtenues et les faire évoluer par suite jusqu'à ce qu'on arrive à un état de stagnation.

Codage d'une population :

Le premier pas dans l'implantation des algorithmes génétiques est de créer une population d'individus initiaux où chaque individu doit être codé sous forme de chromosomes. Le codage ne suit pas une loi particulière, il peut être une suite de bits, un chemin (permutation), un ensemble de valeurs, un arbre ..., autrement dit il va dépendre essentiellement du type du problème à résoudre. La figure 3.9 illustre les différents types de codage rencontrés dans les algorithmes génétiques.

Chromosome A	101100101100101011100101
Chromosome B	111111100000110000011111

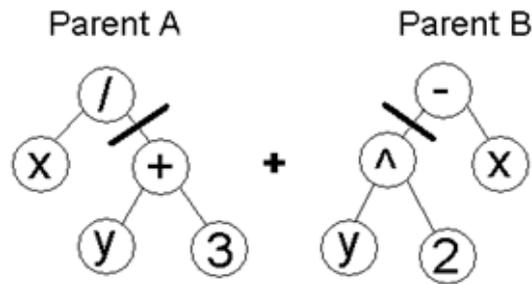
Codage binaire

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

Codage à base de permutation

Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(back), (back), (right), (forward), (left)

Codage à base de valeurs



Codage à base d'arbre

Figure 3.9:Formes de codage [52]

Fonction d'évaluation (fitness) :

Afin de mesurer la qualité des individus, l'algorithme génétique utilise une fonction appelée "fitness". Cette fonction va permettre de distinguer les meilleures solutions et de les préserver par suite [53].

Croisement :

Le croisement est l'opération qui permet de coupler les individus de la même population afin de l'enrichir avec de nouveaux membres, on distingue quatre types de croisement:

- **Croisement en un point:** comme son nom l'indique ce type de croisement s'effectue par rapport à un point aléatoirement choisi. La figure suivante montre le croisement en un point.

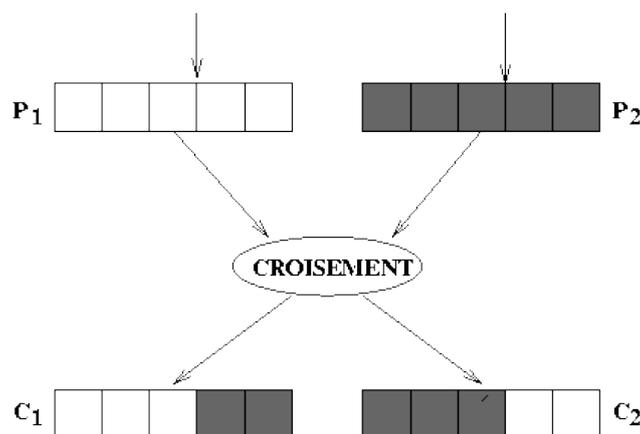


Figure 3.10:Croisement en un point [52]

- **Croisement en deux points:** dans ce type de croisement, le chromosome sera découpé en deux parties, le couplage s'effectue par suite en prenant deux parties du premier parent et une partie du deuxième et vice versa.

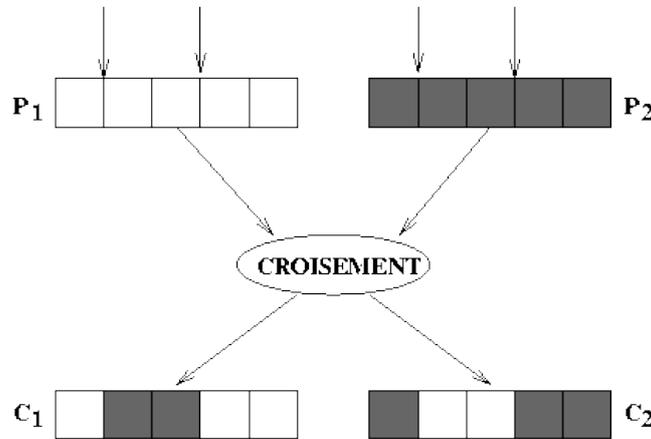


Figure 3.11: Croisement en deux points [52]

- **Croisement uniforme:** le troisième type de croisement s'effectue selon un masque prédéfini. Le principe est de choisir une gène du premier parent si elle correspond à un sinon une gène du deuxième parent si ça correspond à zéro. Le deuxième fils sera généré inversement ("en cas du zéro on choisit une gène du premier parent sinon une gène du deuxième parent").

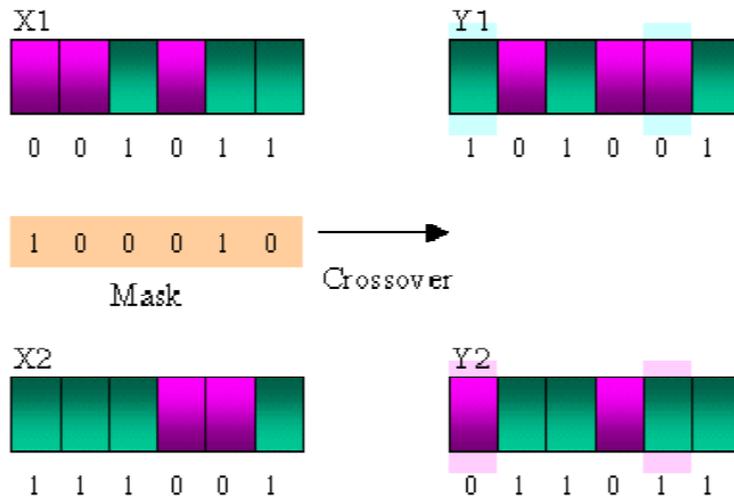


Figure 3.12: Croisement uniforme [53]

- **Croisement arithmétique:** l'idée est de générer un ensemble de fils à travers une opération quelconque.

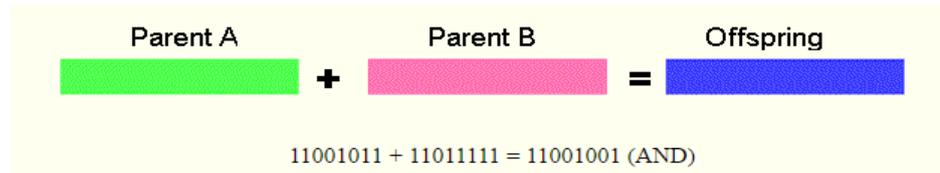


Figure 3.13: Croisement arithmétique [52]

Mutation :

Afin d'explorer la totalité de l'espace de solutions, l'opération de mutation est considérée comme l'une des principales étapes du déroulement de l'algorithme génétique. Le principe de la mutation consiste à tirer aléatoirement une gène et la remplacer par une autre valeur choisie soit aléatoirement soit au le voisinage de la solution générée. La figure 3.11 montre le processus de mutation.

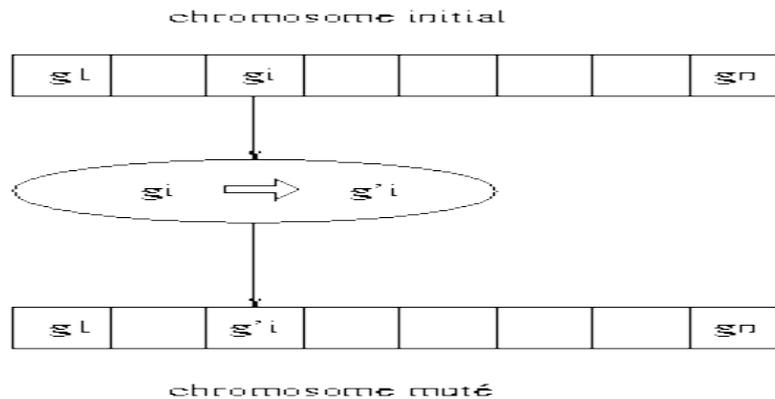


Figure 3.14: Illustration d'une mutation [52]

La sélection :

La sélection est l'une des étapes les plus essentielles lors du déroulement de l'algorithme génétique. Elle consiste à conserver les meilleurs individus pour qu'ils contribuent à la construction de la prochaine génération. On distingue plusieurs façons de sélections:

- **Sélection par roulette :** l'ensemble des individus seront représentés par une roulette, où chacun d'entre eux lui sera attribué une zone avec une taille relative à sa qualité parmi les membres de sa population.

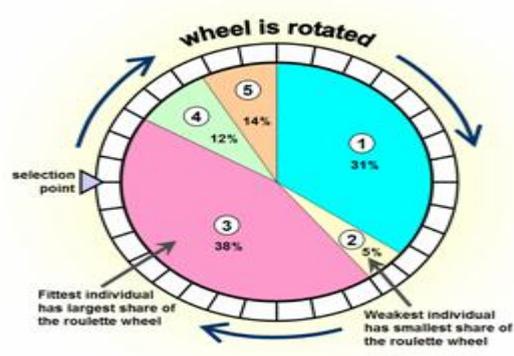


Figure 3.15: Sélection par roulette [52]

- **Sélection par rang:** la sélection par rang a été introduite afin de remédier à la convergence rapide de la sélection par roulette. Le principe de cette sélection est d'associer à chaque individu un rang, ensuite la sélection se fait selon une probabilité relative au rang calculé et non pas selon la qualité des individus.

Chromosomes	1	2	3	4	5	6	Total
Probabilités initiales	89 %	5 %	1 %	4 %	3 %	2 %	100 %
Rang	6	5	1	4	3	2	21
Probabilités finales	29 %	24 %	5 %	19 %	14 %	9 %	9 %

Exemples de sélection par rang pour 6 chromosomes.

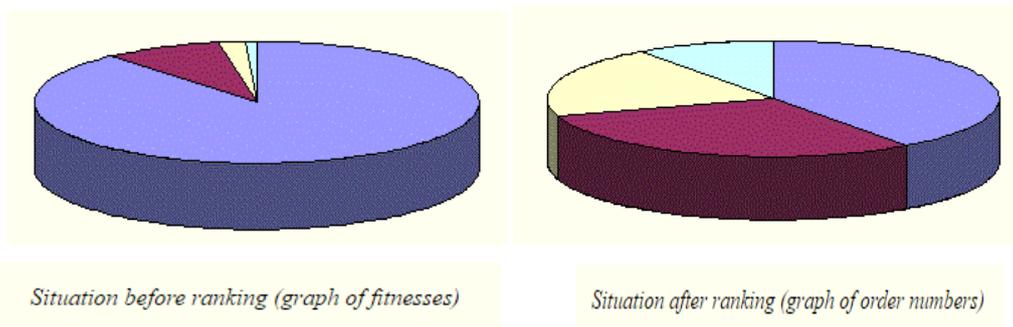


Figure 3.16: Sélection par rang [52]

- **Sélection "steady state":** le principe de cette sélection est de faire survivre une partie de la population initiale et de remplacer les individus les plus faibles par d'autre de la nouvelle génération des fils.

- **Sélection par tournoi:** l'idée est de construire un ensemble de paires d'individus et d'organiser un tournoi entre les composants de chaque paire. La règle générale de la compétition consiste à définir une probabilité de victoire pour chaque individu ("le meilleur individu doit avoir une grande probabilité (entre 70% et 100%) afin d'augmenter ses chances pour emporter le tournoi). La figure montre un exemple de sélection par tournoi.

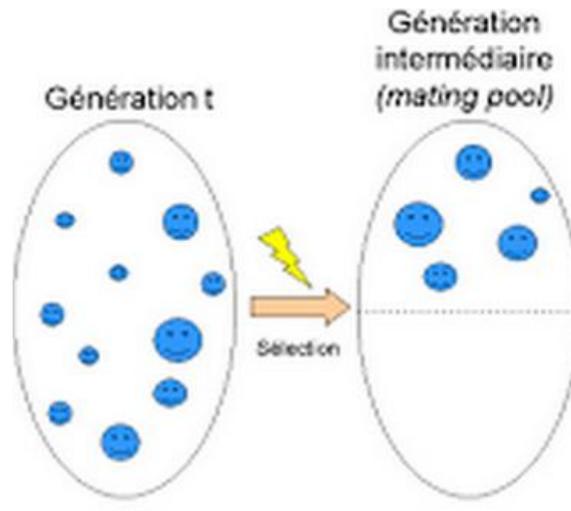


Figure 3.17: Sélection par tournoi

- **La méthode d'élitisme:** A la création d'une nouvelle population, les opérations de croisement et de mutation augmentent la chance de perdre les meilleurs individus durant le déroulement de l'algorithme. La méthode d'élitisme permet d'éviter cela où on copie un ou plusieurs meilleurs individus dans la nouvelle génération. Ensuite, on génère le reste de la population selon l'algorithme de reproduction usuel.

Implémentation d'un algorithme génétique :

La mise en place d'un algorithme génétique peut suivre l'organigramme de la figure suivante.

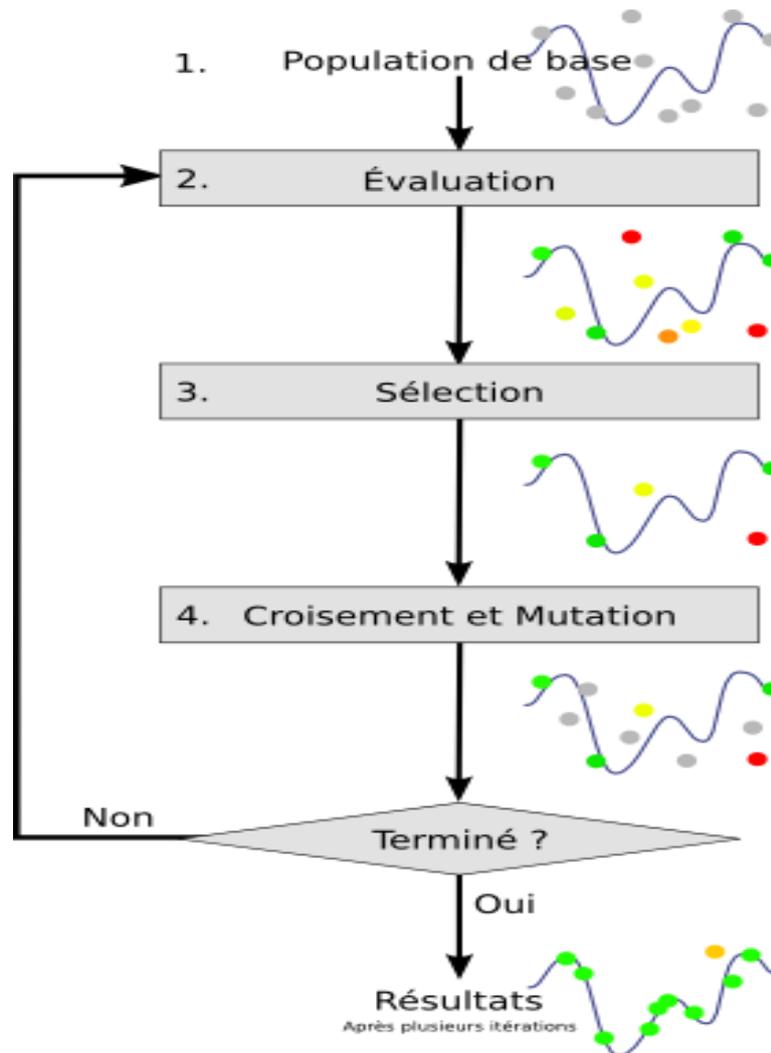


Figure 3.18: Organigramme d'un algorithme génétique

3.4. Conclusion

Dans ce chapitre, nous avons exploré les différentes techniques pour traiter les problèmes NP_difficiles et nous avons mis l'accent sur les méta-heuristiques en particulier.

Cette exploration nous a permis de tirer profit de chacune d'elles et nous a permis également de justifier notre choix pour traiter le problème de routage dans les RCSF à grande échelle.

4. Schémas de routage basés sur les algorithmes génétiques

4.1. Introduction

Les protocoles de routage classiques ont prouvé leurs limites quand le nombre de nœuds augmente dans le réseau ou quand les réseaux sont déployés dans des environnements bruités. Pour faire face à ces deux situations les chercheurs ont optés pour les heuristiques et l'intelligence artificielle pour préserver les performances des protocoles de routage classiques dans ces deux cas de figure.

Dans cette optique, nous avons développé deux schémas de routage pour surpasser le problème du passage à l'échelle dans les RCSF. Le premier schéma de routage proposé est basé sur les algorithmes de routage alors que le deuxième est une amélioration du premier dans lequel nous avons introduit la métrique "ajustement de la portée de transmission".

Dans ce chapitre, nous présentons les deux schémas de routage que nous avons développés et les outils nécessaires pour les mises en place tels que la projection des algorithmes génétiques sur le routage dans les RCSF et le simulateur "OMNET++" comme outil de développement.

4.2. Environnement du développement Omnet++

Omnet++ est une application de base (Framework) orientée objet qui permet de définir des scénarios de simulations en utilisant l'ensemble d'outils et de modules qu'il offre. La figure 4.1 montre la structure générale d'un réseau sur Omnet++.

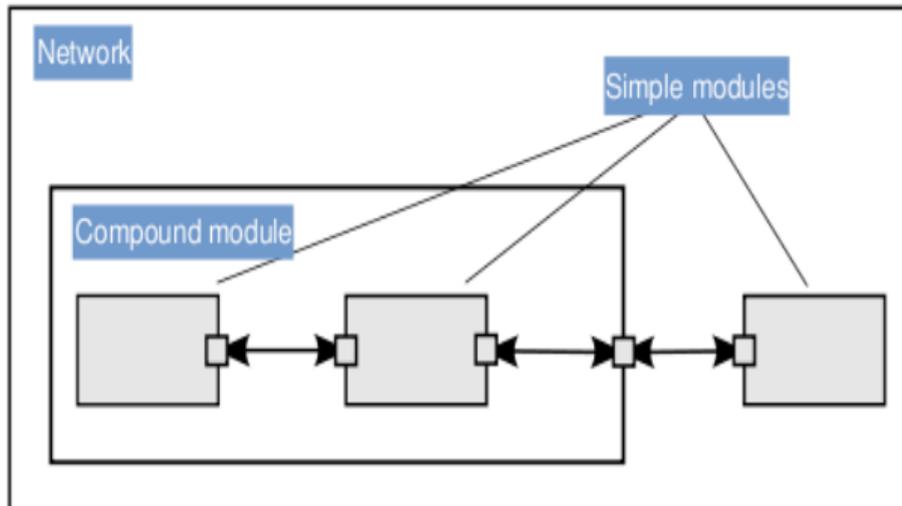


Figure 4.1: Structure d'un réseau sur Omnet++ [54]

Afin de pouvoir développer avec Omnet++ il est nécessaire de savoir manipuler les composants de base afin de les utiliser dans une simulation :

4.2.1. Les modules

Un module dans Omnet++ [54] est un composant graphique représenté par une classe `cModule` écrite en c++. Il est important de savoir qu'un module peut être redéfini par l'utilisateur afin de réaliser le comportement souhaité lors de la simulation.

La partie active d'une simulation est représentée par ce qu'on appelle un module simple et donc c'est à ce niveau où un utilisateur doit intervenir et faire les changements requis. En outre, un module peut être aussi composé de plusieurs modules simple.

```
// file: HelloModule.cc
#include <omnetpp.h>
using namespace omnetpp;

class HelloModule : public cSimpleModule
{
protected:
    virtual void initialize();
    virtual void handleMessage(cMessage *msg);
};

// register module class with OMNeT++
Define_Module(HelloModule);

void HelloModule::initialize()
{
}

void HelloModule::handleMessage(cMessage *msg)
{
}
```

Figure 4.2: Exemple d'un module Omnet++ [54]

Chaque classe qui représente un module simple contient les méthodes suivantes [54]:

- La méthode `initialize()`: cette méthode est invoquée juste après le lancement de la simulation et l'instanciation des composants du réseau. Comme son nom l'indique elle est généralement utilisée pour initialiser l'ensemble des variables ou bien pour planifier des tâches à effectuer par le module lui-même
- La méthode `handleMessage()`: cette méthode sera invoquée chaque fois que le module reçoit un message et donc la redéfinition de cette méthode va permettre de définir le comportement du module lorsqu'un événement de réception aura lieu.

4.2.2. Les canaux de communication(Channel)

Les canaux de communication sont représentés par la classe `cChannel` dont les composants gèrent la partie communication entre les modules. Nous distinguons trois types de canaux de communication [54]:

- `cIdealChannel`: la transmission dans ce type de canal est instantanée et donc les temps de transmission et de propagation sont nuls.
- `cDelayChannel`: en cas d'utilisation de ce type de canal le développeur peut définir un temps de propagation lors de l'acheminement de l'envoi d'un message.
- `cDatarateChannel`: un canal qui gère le délai de propagation, le taux d'erreur et le débit de données.

Il est à noter qu'il est possible qu'un utilisateur pourra définir son propre canal de communication. La figure 4.3 illustre la hiérarchie des objets sous Omnet++.

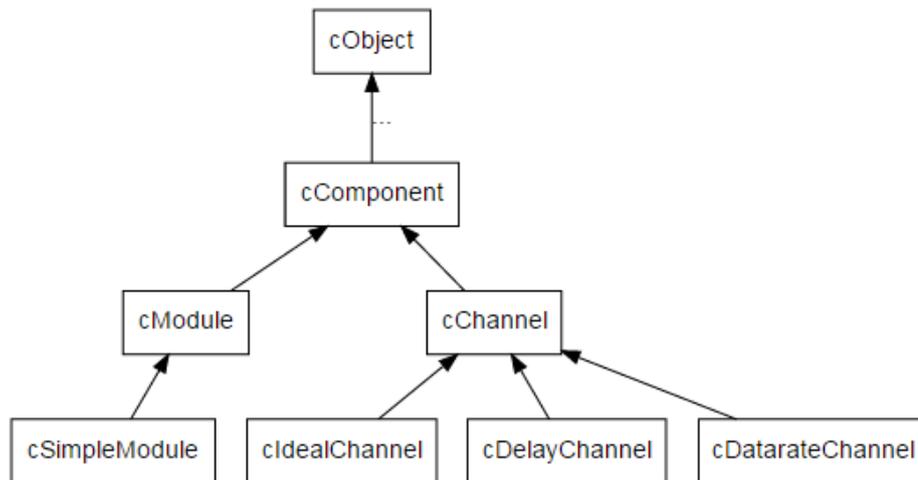


Figure 4.3: Hiérarchie des objets sous Omnet++ [54]

4.2.3. Les messages

Les messages représentent le moyen de communication entre les acteurs du réseau et même au sein d'un module. En plus des champs prédéfinis dans un message (la source, la destination, le type du message, le nom du message, sa priorité, son estampille..) un utilisateur peut définir d'autres en héritant de la classe ci-dessus.

La manipulation des champs d'un message peut s'effectuer à travers les méthodes mutateurs comme l'indique la figure suivante.

```

void setName(const char *name);
void setKind(short k);
void setTimestamp();
void setTimestamp(simtime_t t);
void setSchedulingPriority(short p);
  
```

Figure 4.4: Quelques méthodes utiles pour manipuler un message [54]

4.2.4. Les fichiers de descriptions "Ned File"

L'utilisateur peut décrire la structure de son réseau à travers le langage NED où il peut définir les modules composant son réseau, les connexions et les paramètres de ce dernier. Un fichier de description permet aussi de définir de nouveaux modules ou canaux de communications et les associer à d'autres classes codées en c++ qui vont se charger de leur gestion [54].

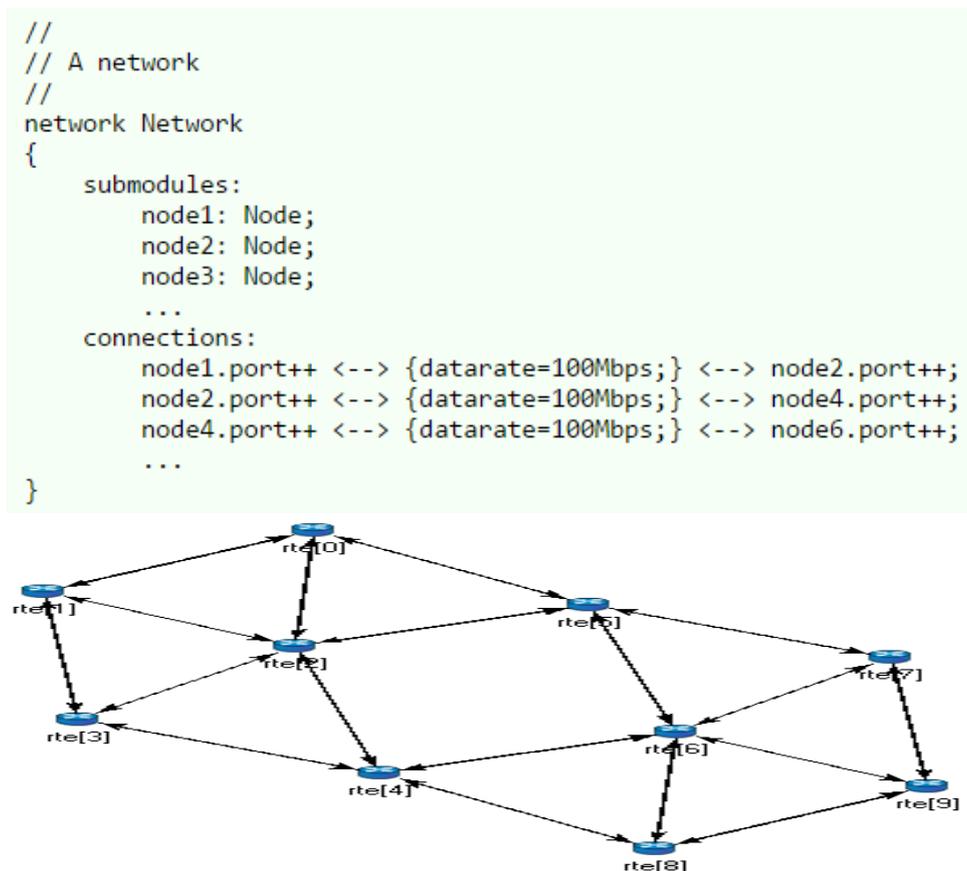


Figure 4.5: Définition d'un réseau [54]

4.3. Schéma de routage basé sur les algorithmes génétiques

Comme nous l'avons cité dans l'introduction, l'objectif de ce projet de fin d'études était la mise en place d'un algorithme de routage basé sur les algorithmes génétiques. A cet effet, pour atteindre cet objectif, nous avons procédé selon la démarche suivante:

4.3.1. La désignation de la population initiale

La constitution de la population initiale est une étape critique pour la continuation de l'algorithme de routage, où plusieurs critères doivent être satisfaits tels que la réalisabilité des individus proposés (représentent des chemins corrects qui mènent vers la station de base) et leurs qualités aussi où il est préférable de commencer avec une population prometteuse. Le choix des chemins initiaux dans l'algorithme proposé est assuré par les méthodes suivantes:

- **La méthode find_Route()** : cette méthode est chargée de trouver une route vers la station de base. Par ailleurs, le particulier dans les chemins trouvés par cette procédure de

recherche de chemins c'est qu'ils sont constitués de nœuds capteurs qui se situent l'un devant l'autre comme suite.

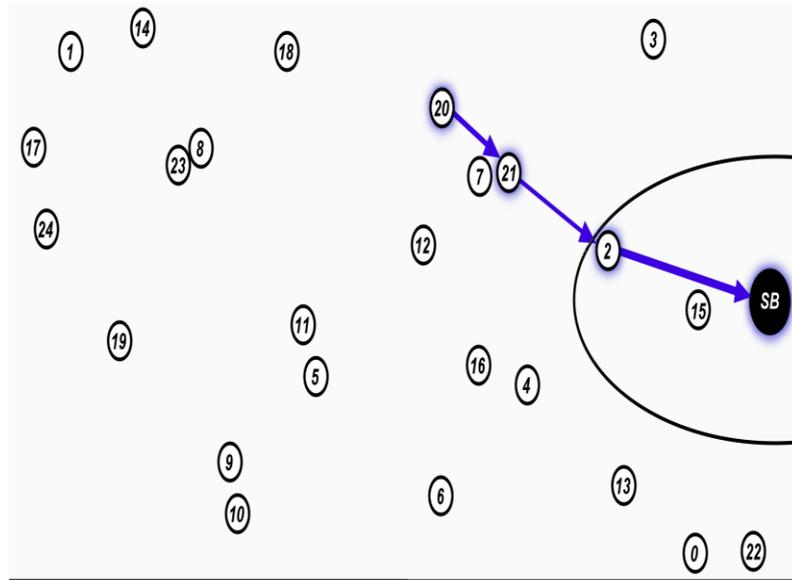


Figure 4.6: Illustration du fonctionnement de find_Route()

Le principe général de cette méthode de recherche de chemins consiste à choisir le premier nœud rencontré et qui respecte le critère énergie (il a suffisamment d'énergie pour envoyer un paquet) et le critère position (il se situe devant le nœud possédant le paquet i.e. il se trouve dans la direction de la station de base) afin de l'ajouter au chemin. Le nœud désigné à son tour va refaire les mêmes étapes jusqu'à ce qu'on atteigne la station de base. En cas d'échec la méthode find_Route supprime le dernier nœud appartenant au chemin et l'insère dans une liste des nœuds éliminés, puis elle explore d'autres voisins comme suite.

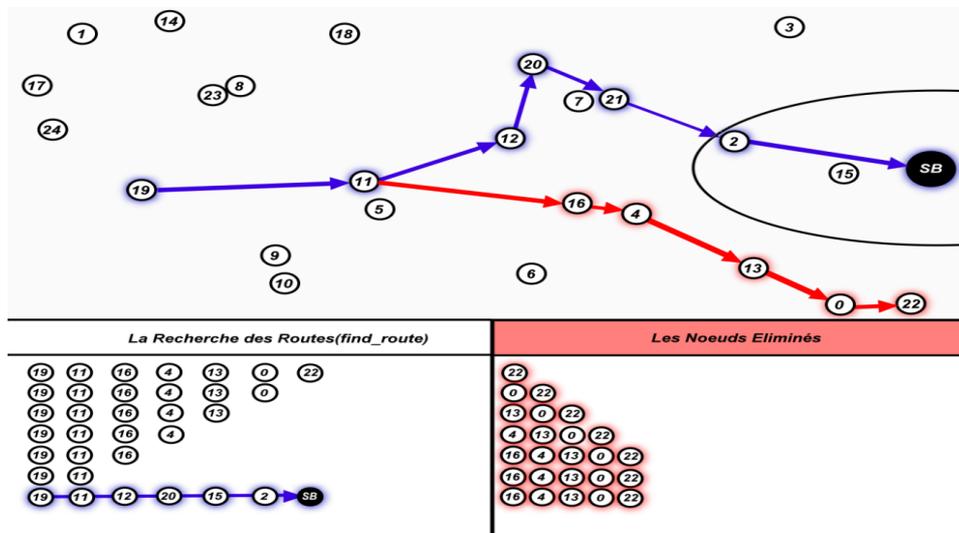


Figure 4.7: Illustration du fonctionnement de find_Route()

Enfin, la méthode sera face à deux situations:

- Un chemin valide et donc la méthode retourne un succès.
- Un chemin constitué seulement du nœud source ce qui veut dire que la méthode n'a pas pu trouver une route qui satisfait les critères énumérés ci-dessus, dans ce cas on fait appelle à la méthode **find_Enhanced()**.

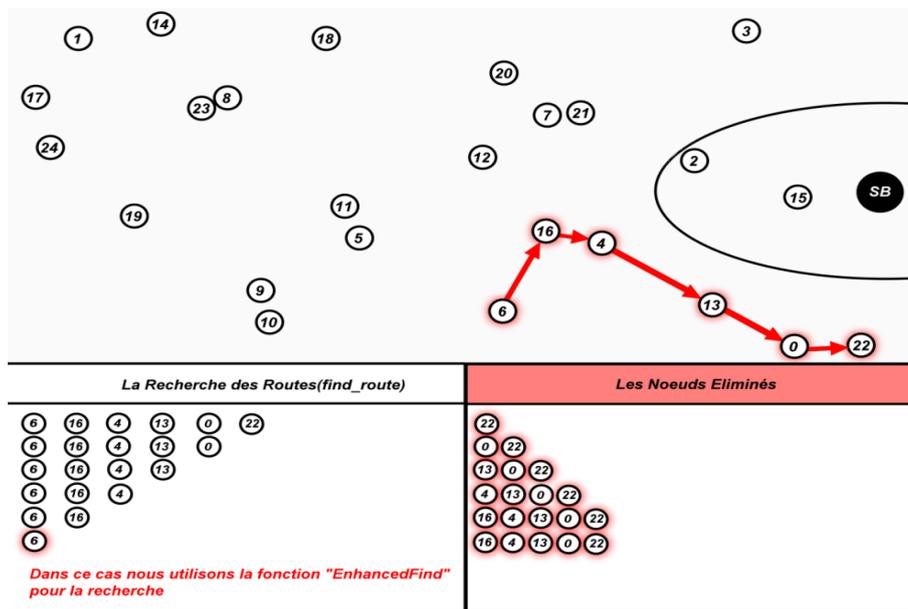


Figure 4.8: Cas d'utilisation de find_Enhanced()

```

Find_Route(source,liste des nœuds constituant le chemin,tableau de position des nœuds « Node[] »)
Début
  Initialisé la liste des nœuds éliminé
  Entier : last=source
  Si (voisin_station(last) ) alors critère d'arrêt=vrai
  Tantque( non critère d'arrêt) faire
    Debut
      Entier : indice =random (entre 0 et nombre des nœuds)
      Pour (i=0 jusqu'au nombre des nœuds-1) faire
        Debut
          Si (distance(station,indice mod (nombre des nœuds))< distance(station,last)) et
            (voisin(last, indice mod (nombre des nœuds)))
          alors
            debut
              si (Node[indice mod (nombre des nœuds)].Energie <energie equivalente a un envoi)  alors
                continue
            sinon
              debut
                ajouter (indice mod nombre des nœuds) dans liste
                last=indice mod nombre des nœuds
              Fin
            finsi
          Fin
        finpour
      si (i>nombre des nœuds) alors
        Debut
          si (liste contient que le nœuds source) alors
            si (appeler find_Enhanced =0) alors aucune route trouvée
              sinon critère d'arrêt=vrai
          finsi
        sinon
          Debut
            ajouter le dernier nœud de liste dans la liste elemine
            supprimer le dernier nœud de liste
          Fin
        finsi
      fintantque
    Fin
  Fin

```

Figure 4.9:pseudo-code de la procédure find_Route

Dans la méthode find_Enhanced() on accepte de faire un pas vers l'arrière et de lancer une recherche de chemin menant vers l'avant jusqu'à ce qu'on arrive à la station de base, en cas d'échec on fait encore un pas vers l'arrière et on relance la même procédure de recherche. Le particulier dans cette méthode c'est si il existe un chemin menant vers la destination finale il va être trouvé sinon le nœud va être considéré isolé et par suite tous paquets qui lui arrive sera considéré comme étant un paquet perdu.

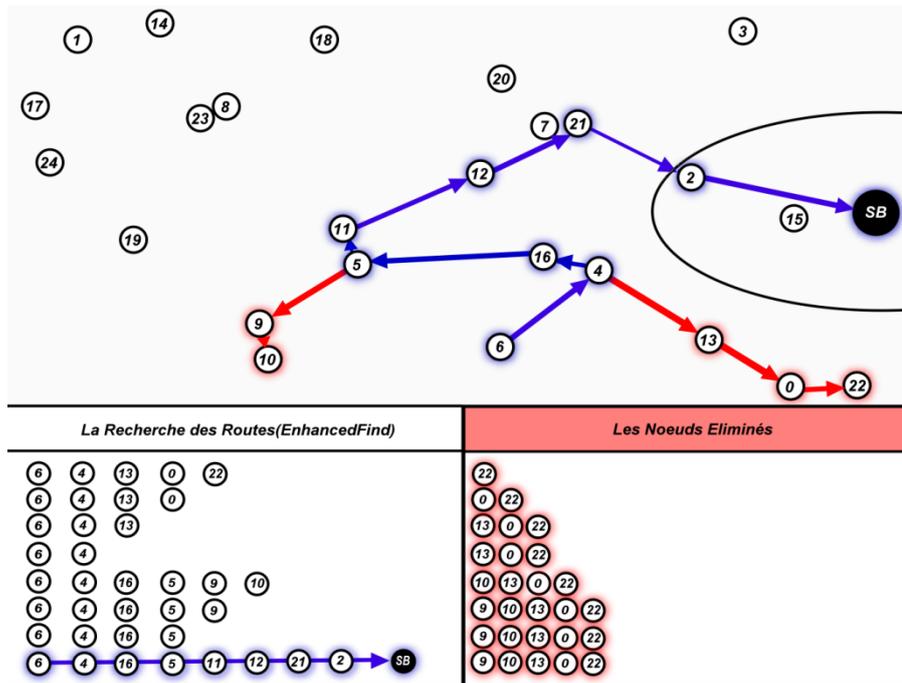


Figure 4.10: Illustration du fonctionnement de find_Enhanced()

Find_Enhanced(source,liste:liste originale passée par la fonction qui fait l'appel find_Enhanced,
tableau de position "Node",
list_ext:une liste utilisé pour éviter les cycles,liste elemine
)

Début

Entier : indice=random(entre 0 et nombre des noeuds)

Pour (i=0 jusqu'au nombre des nœuds-1) faire

 indice=indice mod nombre des noeuds

 Si(exist(liste elemine,indice)) alors continue

 Si(exist(list_ext,indice)) alors continue

 Si (exist(liste,indice!=1) et(indice!=source) et (voisin(indice,src)) alors

 Debut

 si (Node[indice].Energie < energie equivalente a un envoi) alors continue

 si (Find_F(indice,liste)=1) retourner 1

 sinon

 Debut

 ajouter(indice,l)

 ajouter(indice,liste elemine)

 ajouter(indice,list_ext)

 Fin

 finsi

 Si (find_Enhanced(indice,l,list_ext,liste elemine) alors

 Debut

 fusionner(list,l)

 retourner 1

 Fin

 finsi

supprimer(list_ext,indice)

finpour

retourner 0

Fin

Figure 4.11:pseudo-code de la fonction find_Enhanced

En outre il existe une autre méthode, appelée la méthode `find_Enhanced_F()`. Cette méthode est utilisée par la méthode précédente et elle est chargée de trouver un chemin direct vers la station de base comme dans la première méthode la seule différence c'est qu'en cas d'échec cette méthode retourne une valeur nulle. Il faut citer aussi que cette méthode évite de choisir les nœuds déjà explorés par les méthodes précédentes et qui ont mené à un échec (à travers la liste des nœuds éliminés).

L'algorithme de routage proposé fait appel à ces méthodes quatre fois de suite ce qui nous donne une population initiale avec quatre chemins différents ou bien identiques en cas où il n'existe qu'un seul chemin qui mène vers la station de base. L'occurrence de ce dernier cas est très rare puisqu'on a visé des réseaux denses.

4.3.2. La fonction objective "fitness"

L'évaluation des individus s'effectue selon le type de l'algorithme traité. On distingue deux évaluations :

- une évaluation selon la quantité d'énergie épuisée par toute la population
- une évaluation selon la distance des chemins constituant la population.

4.3.3. L'étape de croisement

Dans cette étape l'algorithme de routage localise un point de croisement situé dans les dernières 30% du chemin du parent le plus court (court en terme de longueur i.e.: le nombre des nœuds composant le chemin). Puis, nous essayons ensuite de générer tous les fils possibles en couplant les quatre parents. Après que le croisement sera effectué une vérification et une correction (si c'est nécessaire) des chemins résultant aura lieu afin de garantir la réalisabilité de la solution finale par suite. La figure montre comment se fait le croisement par l'algorithme proposé.

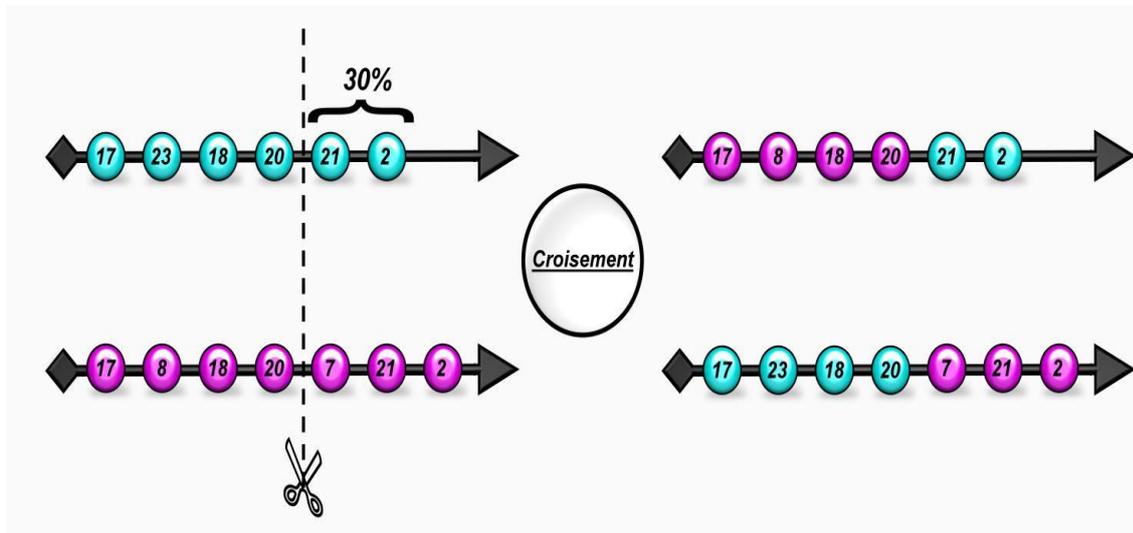


Figure 4.12: Etape de croisement

4.3.4. L'étape de sélection

La sélection effectuée lors du déroulement de l'algorithme est de type "steady-state" où on garde les meilleurs deux parents et on remplace les parents les plus faibles par les deux meilleurs des douze fils générés. Cette sélection va nous permettre de garder que les meilleurs individus et de les combiner afin de tirer d'autres qui sont encore meilleurs et de converger rapidement vers une solution finale.

Par ailleurs, le choix du meilleur peut se baser sur des critères différents, par exemple dans le cas de la version de l'algorithme de routage qui favorise l'énergie le critère de sélection du meilleur dépend de la quantité d'énergie que possède l'ensemble des nœuds constituant le chemin contrairement à la version qui favorise le plus court chemin où le choix se fait selon la distance parcourue dans chaque route trouvée.

4.3.5. Critère d'arrêt

Comme tout algorithme génétique le critère d'arrêt correspond à un état de stagnation, dans notre cas on arrête la procédure de création d'une nouvelle population dès qu'on arrive plus à améliorer la solution. Or le moyen de comparaison entre les générations s'effectue selon la version de l'algorithme en cours d'exécution. En cas de favorisation d'énergie on cumule le coût d'énergie épuisée par toutes la population et on le compare avec l'énergie épuisée par la nouvelle génération on continue ce processus tant qu'on obtienne des individus qui épuisent moins d'énergie. La version distance fait pareil mais avec le critère longueur où

l'algorithme de création continue à se dérouler tant qu'il trouve des individus qui sont meilleurs en termes de distance cumulée.

4.3.6. Choix de la meilleure solution

Dans la version énergie de l'algorithme le choix du meilleur individu est basé sur la maximisation du rapport entre la quantité d'énergie possédée par l'ensemble d'individus constituant le chemin et la quantité d'énergie dissipée le long du chemin i.e. :

$$\text{Max}_{i=1,\dots,4} \left(\frac{\text{Energie}_{\text{Totale}}(Ch_i)}{\text{Energie}_{\text{Dissipée}}(Ch_i)} \right)$$

Dans la version distance de l'algorithme, le choix du meilleur individu s'effectue selon la longueur du chemin parcouru, où on sélectionne la route la plus courte comme étant le chemin de routage le plus adapté.

La figure 4.13 illustre le fonctionnement de l'algorithme de routage que nous avons proposé.

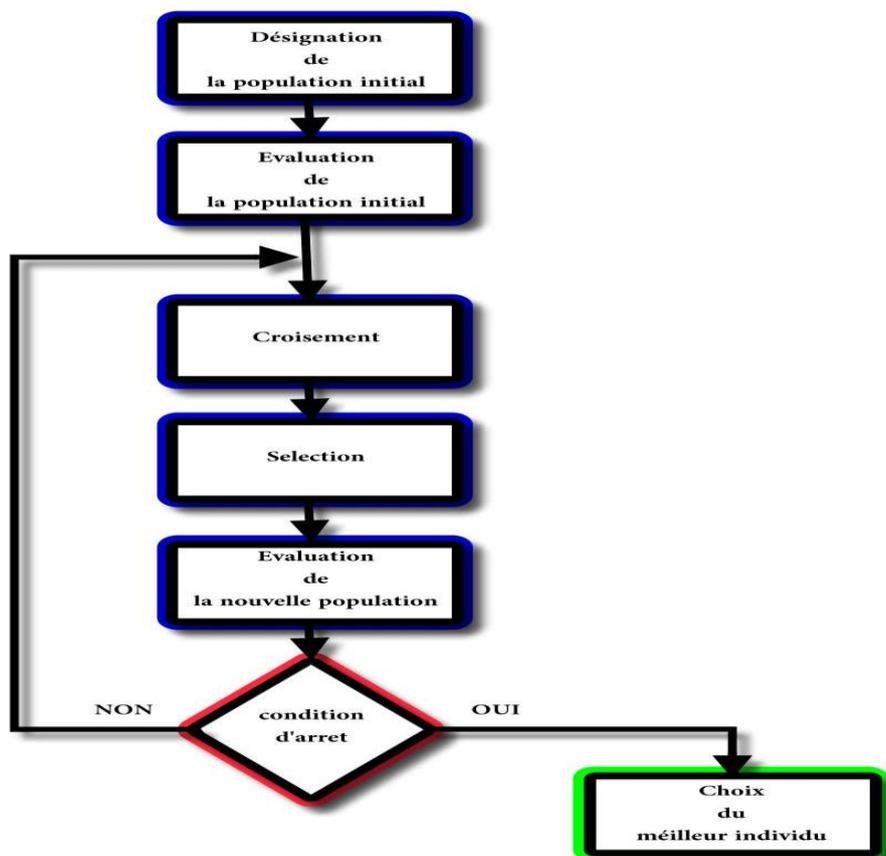


Figure 4.13: Organigramme de l'algorithme proposé

4.4. Implémentation sur Omnet++

La première étape de l'implémentation de l'algorithme de routage est la définition des messages et des modules suivants:

4.4.1. Le module station de base

Le module station de base est un module simple associé à la classe "basestation.cc" et chargé de calculer les chemins pour chaque nœud membre du réseau et de leurs diffuser le prochain saut par suite en exploitant sa connaissance de la topologie. La station de base maintient aussi une table qui estime l'énergie que possède chaque capteur et elle peut la mettre à jour en se basant sur sa connaissance des chemins.

4.4.2. Le module capteur

Le module capteur est un module simple associé à la classe "sensor.cc" qui permet sa gestion et qui définit son comportement en cas d'envoi ou de réception d'un message.

Un module capteur possède les caractéristiques suivantes:

- un ensemble d'interfaces réseau qui permettent l'envoi et la réception des messages.
- une énergie initiale.

La gestion des paramètres précédents est aussi assurée par la classe "sensor.cc", où elle se charge de la mise à jour des valeurs de l'énergie, l'évaluation de l'état du capteur et la mise à jour la table de routage.

Chaque capteur possède une table de routage qui contient le prochain saut à effectuer lors de l'acheminement d'un message.

4.4.3. Définition des types de messages

Le réseau proposé gère trois types de messages définis à l'aide de la classe cMessage et du fichier ".Msg".

- **Message de topologie:** ce type de messages est utilisé lors du lancement du réseau, où chaque capteur explore son voisinage en diffusant un message sur toutes ses interfaces réseau. Par ailleurs, pour chaque message de topologie le capteur récepteur mémorise la source du message et l'interface réseau sur laquelle il la reçut.

- **Message de saut:** ce type de messages est diffusé par la station de base, il contient le nœud destination et son prochain saut. Chaque capteur qui reçoit ce type de message met à jour sa table de routage.
- **Message de données:** comme son nom l'indique ce type de messages est utilisé par chaque nœud voulant émettre des données vers la station de base. Ce message contient la source du message et les données à envoyer.

4.5. Fonctionnement du réseau

Le fonctionnement du réseau peut être décomposé en trois étapes majeures:

- **Etape 1 "découverte":** diffusion des messages de topologie afin que les nœuds puissent s'auto-organiser. La figure 4.14 illustre la phase de découverte.

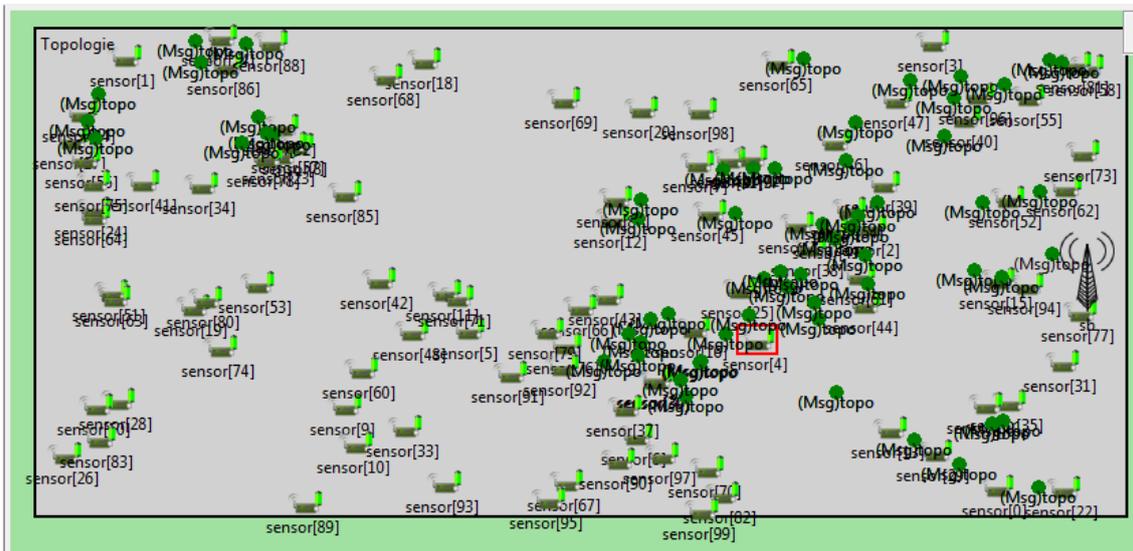


Figure 4.14: Phase de découverte

- **Etape 2 "routage":** cette étape est effectuée par la station de base. Elle consiste à calculer les chemins et à diffuser des messages de sauts par suite. Il faut mentionner que cette étape aura lieu dans les cas suivants:
 - lors de la phase d'initialisation du réseau.
 - pour chaque réception de 100 messages au niveau de la station de base.
 - lors de la détection d'un nœud inactif.

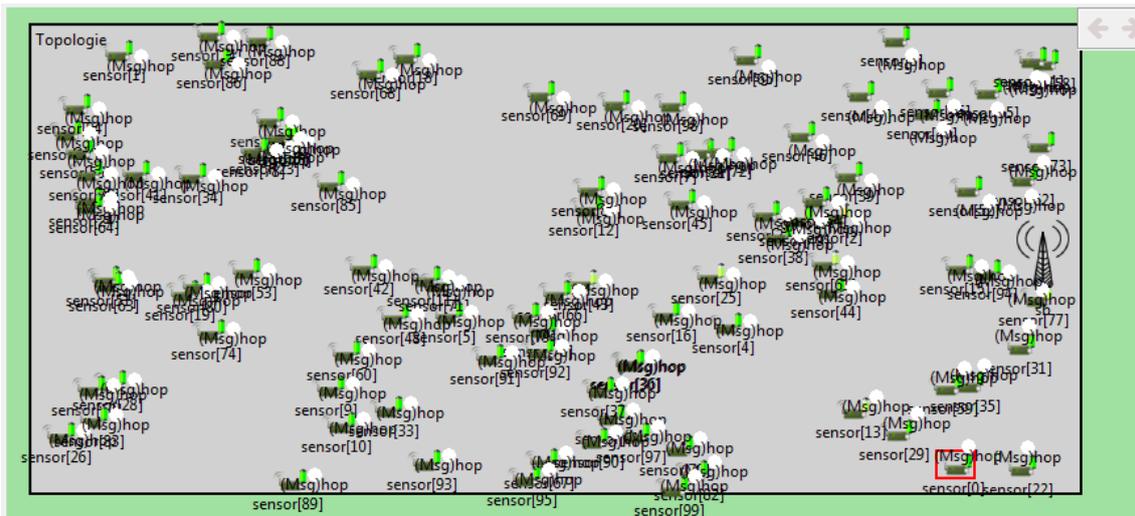


Figure 4.15: Phase d'établissement des routes

- **Etape 3 "échange des données"**: comme son nom l'indique l'échange de données s'effectue dans cette étape en se basant sur les informations de sauts diffusées par la station de base.

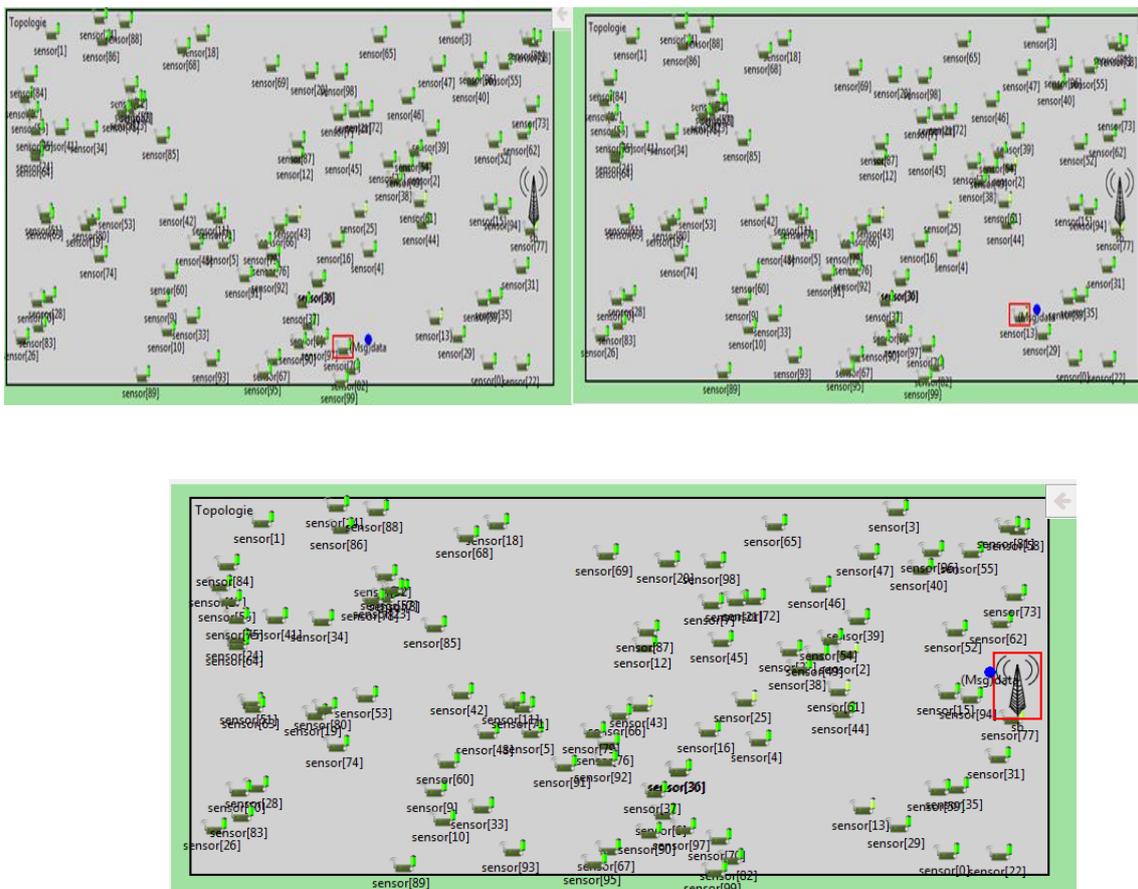


Figure 4.16: Acheminement des données

4.6. Modèle énergétique

Le but de cette simulation est l'évaluation de l'algorithme de routage proposé selon le critère énergétique et donc il était nécessaire de définir un modèle de consommation afin de calculer le coût d'énergie consommée lors d'une transmission. Le modèle choisi correspond à un modèle radio simple, cité dans de plusieurs articles [30,55] et défini comme suit :

- En cas d'envoi :

$$E_{Tx}(l, d) = \begin{cases} l \times E_{elec} + l \times \epsilon_{fs} d^2, & \text{if } d \leq d_0 \\ l \times E_{elec} + l \times \epsilon_{mp} d^4, & \text{if } > d_0 \end{cases}$$

$$d_0 = \sqrt{\epsilon_{fs} / \epsilon_{mp}}$$

- En cas de réception:

$$E_{Rx}(l) = l \times E_{elec}$$

Où :

l: représente la longueur du paquet

Eelec: énergie consommée par le récepteur/transmetteur radio.

ϵ_{fs} : facteur d'affaiblissement du signal quand la distance est inférieure à d_0 .

ϵ_{mp} : facteur d'affaiblissement du signal quand la distance est inférieure à d_0 .

d: représente la distance entre l'émetteur et le récepteur.

4.7. Simulation

4.7.1. Paramètres de simulation

La table présente les paramètres de la simulation.

Paramètres	Valeurs
Zone de simulation	100m X 100m
Energie initiale de chaque capteur	2 Joule
E_{elec}	50 nJ/bit
ϵ_{fs}	10pJ/bit/m ²
ϵ_{mp}	0.0013pJ/bit/m ⁴

Porté de transmission	25 m
Taille du message de saut	232 bits
Taille du paquet de données	4200 bits

Tableau 4-1: Paramètres de la simulation

La simulation s'arrête lorsque 33% des nœuds du réseau deviendront isolés ou inactifs.

4.7.2. Résultats de la simulation

L'évaluation des performances de l'algorithme proposé est faite dans des conditions de simulation différentes non seulement en termes de nombre des nœuds mais aussi dans diverses topologies.

a. La version énergie

L'exécution d'un scénario aléatoirement généré de l'algorithme qui favorise le critère énergie dans le choix des chemins a donné les résultats suivants :

Taille du réseau	Nombre de nœuds directement reliés à la station de base		
	Topologie 1	Topologie 2	Topologie 3
100 nœuds	12 nœuds relais	6 nœuds relais	8 nœuds relais
200 nœuds	17 nœuds relais	21 nœuds relais	18 nœuds relais
300 nœuds	23 nœuds relais	32 nœuds relais	23 nœuds relais
400 nœuds	28 nœuds relais	43 nœuds relais	31 nœuds relais

Tableau 4-2: Description des topologies utilisées

Chaque nœud programme un envoi de données dans un temps choisi aléatoirement et dans l'intervalle $[0,3s]$. L'opération se répète jusqu'à ce que le capteur épuise son énergie ou bien il devient isolé.

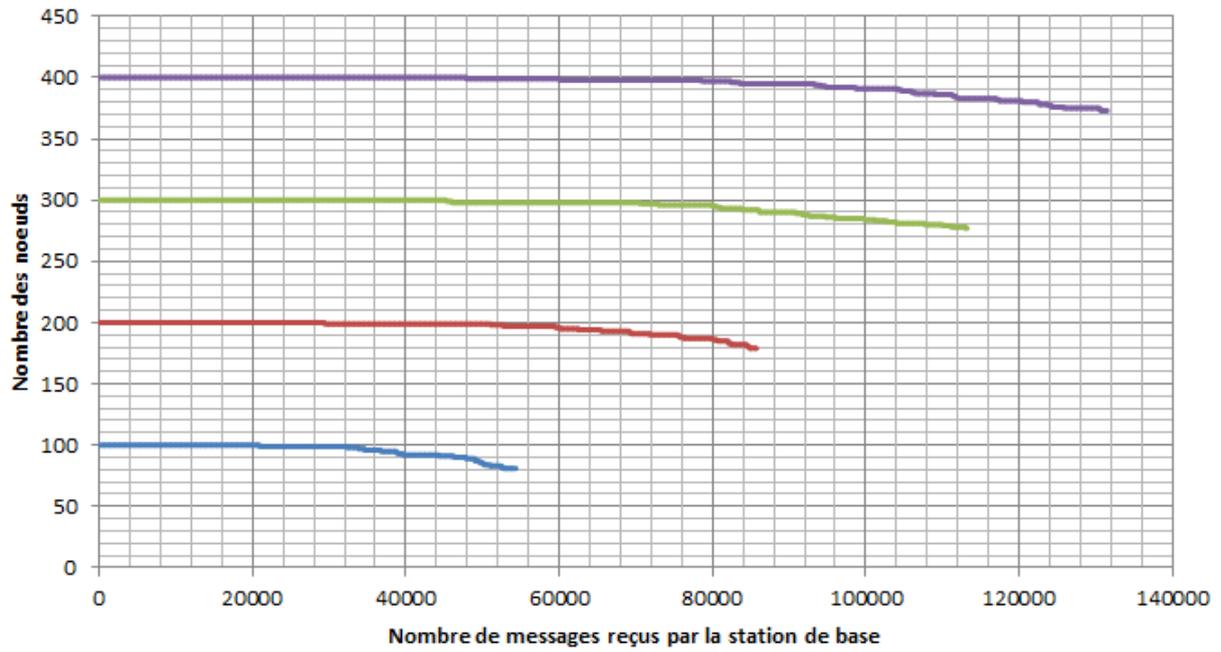


Figure 4.17: Résultats de simulation sur la topologie 1

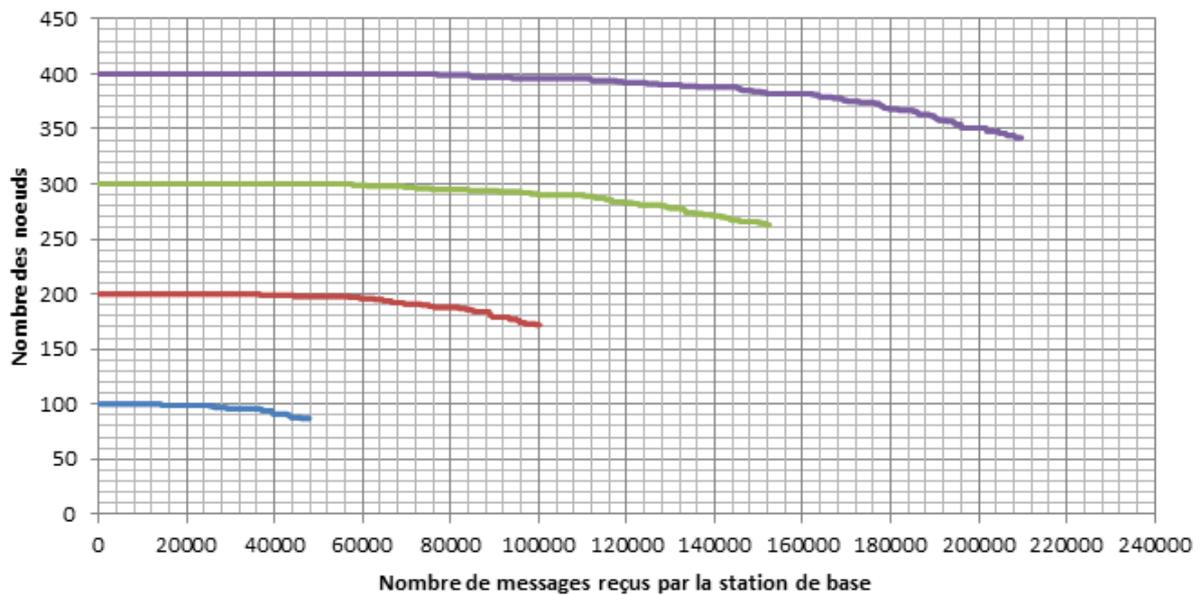


Figure 4.18: Résultats de la simulation sur la topologie 2

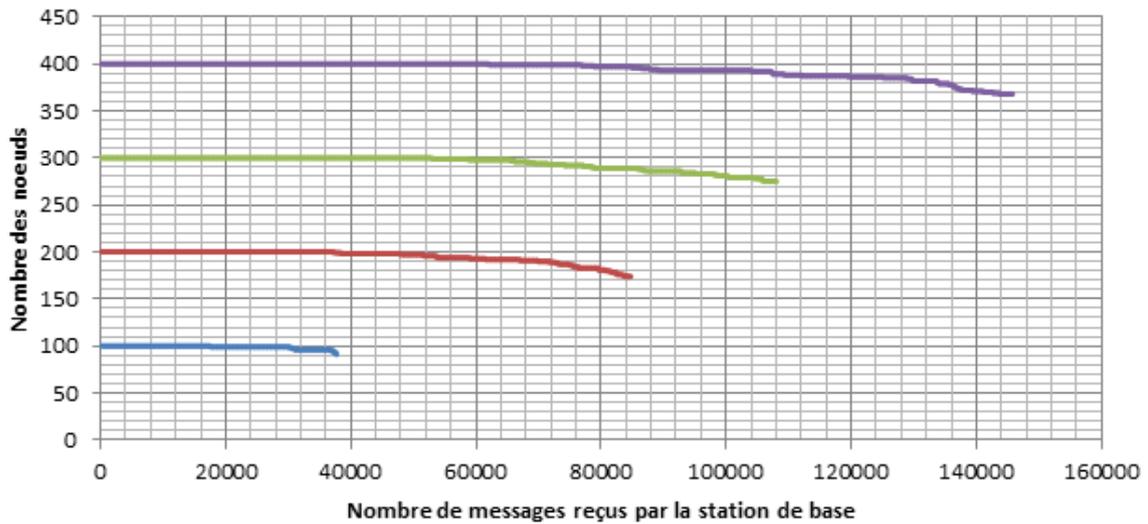


Figure 4.19: Résultats de la simulation sur la topologie 3

b. Version distance

L'exécution d'un scénario aléatoirement généré de l'algorithme qui favorise le critère distance dans le choix des chemins dans la topologie 2 a donné les résultats suivants :

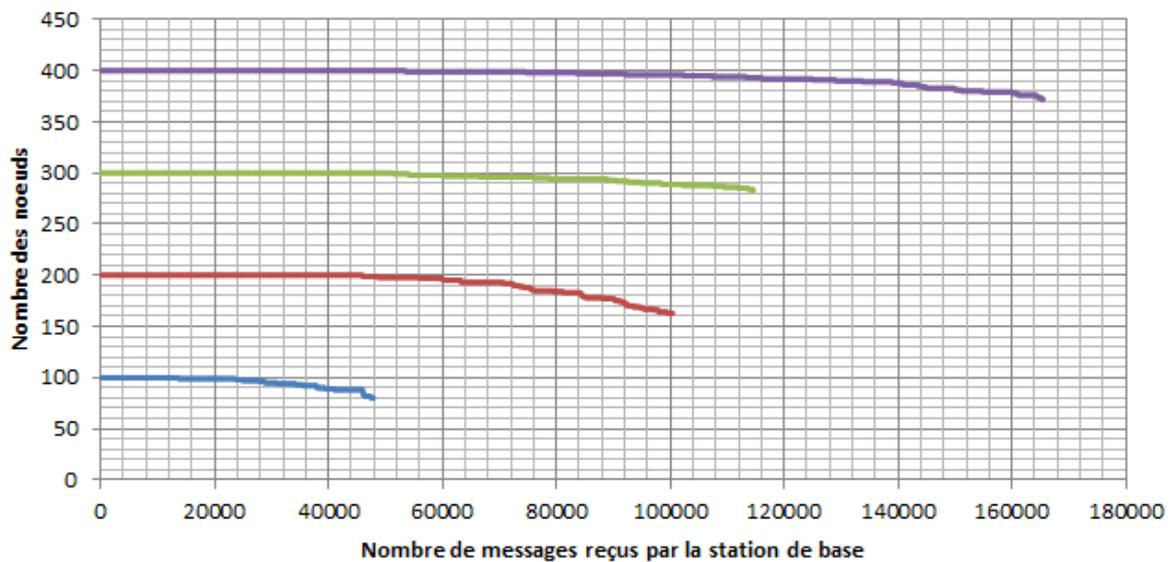


Figure 4.20: Résultat de la version distance de l'algorithme appliqué sur la topologie 2

4.8. Discussion des résultats

Dans les figures précédentes 4. (17, 18, 19) On voit clairement que la répartition des charges sur les différents nœuds avait un effet positif sur la durée de vie du réseau et même sur le nombre des nœuds morts. Par exemple, dans la première topologie dans laquelle il y a 12 relais vers la station de base parmi 100 nœuds présents dans le réseau on a pu avoir environ 55000 messages envoyés avec succès (messages reçus par la station de base) avec seulement 12 nœuds morts. On constate aussi que lors de l'augmentation du nombre des relais le nombre de messages reçus avec succès peut s'améliorer d'avantage. Par exemple dans la deuxième topologie l'ajout de 15 relais parmi 100 nœuds supplémentaire a permis une amélioration de 100% en terme de nombre de messages reçus par la station de base, de même pour la troisième topologie le passage de 200 à 300 nœuds avec l'ajout de 5 relais a permis une amélioration de 30% .

En d'autre part les résultats de simulation de la version distance de l'algorithme proposé étaient très approximatifs à ceux de la version énergie avec une légère amélioration de cette dernière. Ceci pourra être expliqué par le calcul de l'énergie qui se base en premier lieu sur la distance parcourue par le signal transmis.

4.9. Comparaison avec d'autres protocoles :

Comme l'indique la figure 4.21 les résultats obtenus par l'algorithme proposé surpasse ceux de LEACH. On constate que dans LEACH, les nœuds épuisent leurs énergies dans une période très courte alors qu'au même moment l'algorithme proposé recense seulement 9 nœuds morts. En outre, la durée de vie dans LEACH-C est plus ou moins supérieure à celle que de notre algorithme malgré que dans notre algorithme seulement 20 nœuds qui cessent de fonctionner. Ceci est dû à la limitation du rayon de transmission (25 mètres) contrairement au deux autres protocoles où les clusterheads peuvent communiquer directement avec la station de base ce qui permet la continuation de la simulation jusqu'à l'épuisement de tous les nœuds constituant le réseau.

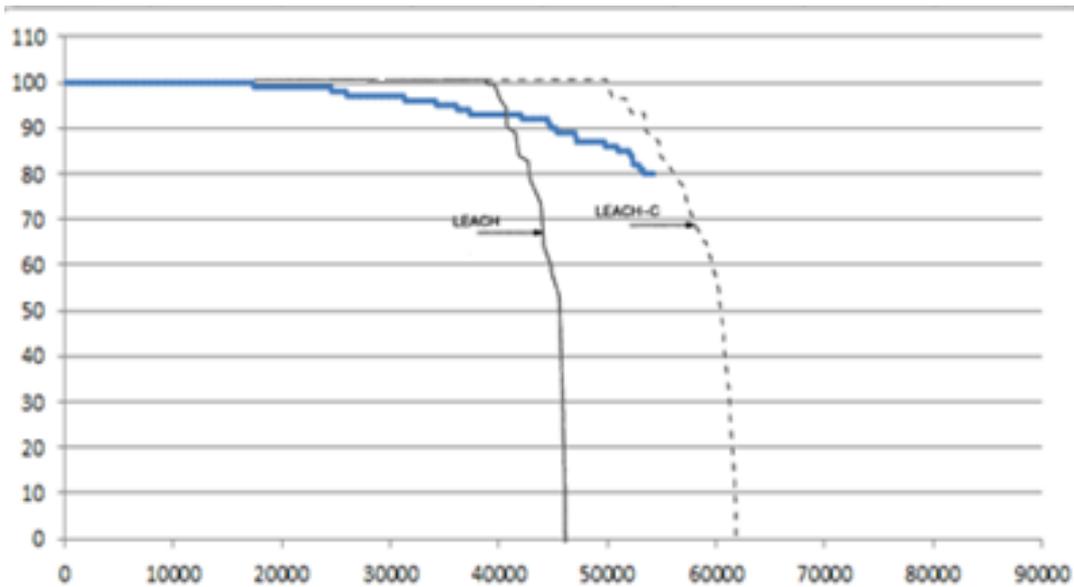


Figure 4.21: Comparaison de la durée de vie de notre algorithme LEACH et LEACH-C

4.10. Améliorations proposées :

Afin d'augmenter les performances de l'algorithme proposé, nous avons opté pour deux améliorations une qui porte sur la méthode de sélection utilisé auparavant, or la deuxième ajuste la puissance de transmission des nœuds les plus proches de la station de base en fonction de l'épuisement d'énergie de ces derniers.

4.10.1. Changement de la méthode de sélection :

Le changement de la méthode de sélection est une des améliorations qui pourront être proposées, où on distingue plusieurs façons de réaliser cette opération comme nous l'avons déjà décrit dans le chapitre précédent.

L'adoption d'un nouveau type de sélection était l'objectif d'une de nos expérimentations. L'aspect de cette nouvelle démarche est similaire à celle de la sélection par tournoi mais la façon est différente, où nous avons organisé une compétition entre l'ensemble des fils générés avec une probabilité de 100% à la faveur du meilleur, une fois l'individu le plus prometteur sera choisi on le retire du groupe des individus en compétition et on l'ajoute dans la nouvelle génération de la population. L'opération sera répétée jusqu'à ce qu'on construise une nouvelle population (i.e. la répétition de cette démarche dépend de la taille de

la population), en d'autres termes cette sélection a pour objectif de choisir les n meilleurs fils générés.

En outre, l'expérimentation a montré que l'introduction de la méthode de sélection proposée a permis d'augmenter le nombre de messages reçus par la station de base tout en préservant la quantité d'énergie consommée par les nœuds où on a constaté que le nombre des nœuds inactifs lors de la fin de la simulation a diminué de 2 à 6 par rapport la version de l'algorithme qui utilise l'ancienne méthode de sélection.

Les figures 4. (22, 23) illustrent les résultats obtenus lors de l'exécution de l'algorithme utilisant la nouvelle méthode de sélection sur la topologie 2 et les comparent à ceux de la première sélection.

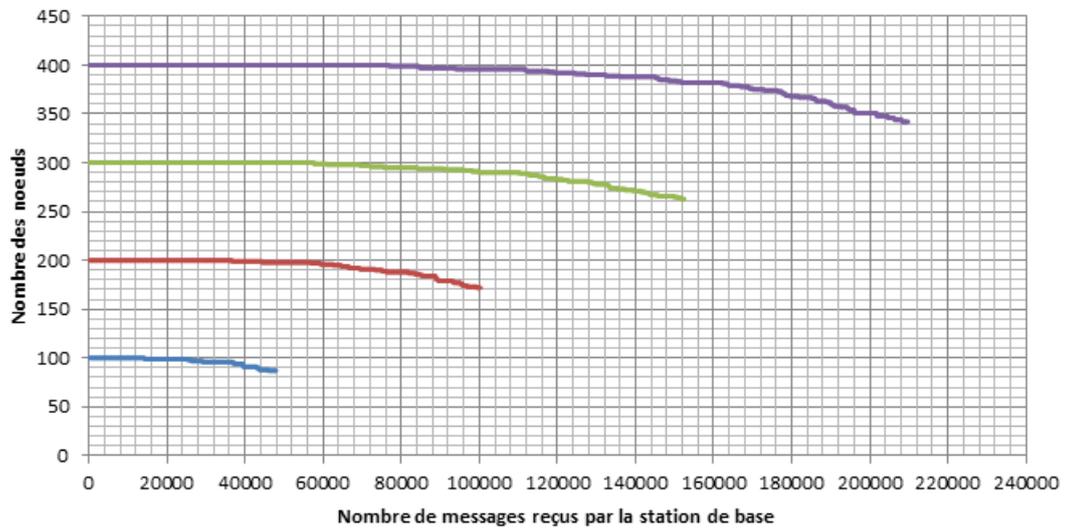


Figure 4.22: Résultat de la simulation avec l'ancienne méthode de sélection

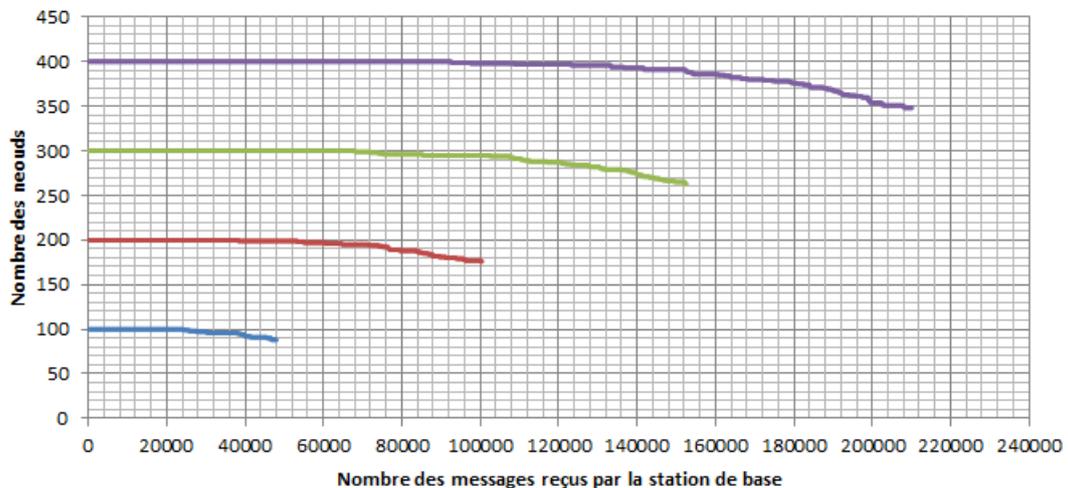


Figure 4.23: Résultat de la simulation avec la nouvelle méthode de sélection

Ci-dessous les résultats obtenus lors de l'exécution des deux versions de la sélection sur cinq différentes topologies.

Taille du réseau (200 Nœuds)	Ancienne méthode de sélection		Nouvelle méthode de sélection	
	Nb de messages envoyés	Nb de nœuds vivants	Nb de messages envoyés	Nb de nœuds vivants
Topologie1	85660	181	85830	182
Topologie2	100253	173	100430	175
Topologie3	84620	175	84847	181
Topologie4	86166	178	86460	181
Topologie5	61573	178	61650	181

Tableau 4-3: Résultats d'exécution des deux méthodes de sélection dans 5 topologies différentes

4.10.2. Routage avec ajustement de puissance de transmission

L'expérimentation a prouvé que la répartition des charges sur les différents nœuds du réseau lors de l'acheminement des données engendre une meilleure conservation d'énergie au sein des capteurs. Cependant, la limitation du rayon d'envoi à 25m affecte énormément la performance de l'algorithme proposé où ça nous pousse à juger que le réseau n'est plus opérationnel malgré que dans toutes les simulations effectuées seulement 20% des nœuds ont épuisés leurs énergies au pire des cas.

Le routage avec ajustement de puissance de transmission est une extension de l'algorithme proposé auparavant, cette démarche se comporte dans un premier temps d'une façon similaire à celle de la version énergie décrite ci-dessus. En revanche une fois que les nœuds qui se situent dans un rayon de 25m de la station de base ont épuisé leurs énergies l'algorithme augmente la puissance de transmission des capteurs les plus proches de la station de base pour qu'ils puissent faire des envois directs vers cette dernière. Ce processus sera répété à chaque fois que les nœuds les plus proches de la destination deviennent inactifs. Cette méthode permet d'exploiter l'énergie de tous les capteurs ce qui mène à un prolongement important de la durée de vie du réseau. La figure 4.24 montre l'amélioration de la durée de vie du premier algorithme proposé en ajustant la puissance d'émission.

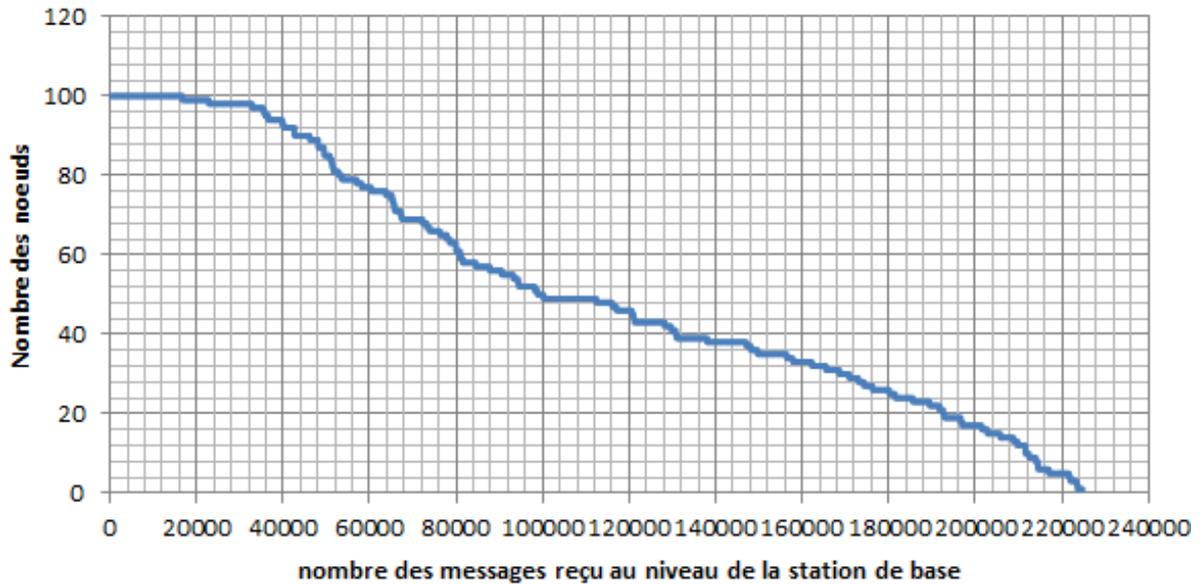


Figure 4.24: résultats de la simulation d'un routage avec ajustement de puissance testé sur la topologie 1

4.11. Conclusion

Dans ce chapitre, nous avons présenté les deux schémas de routages pour les réseaux de capteurs sans fil. Ces deux schémas de routage se basent sur les algorithmes génétiques pour remédier aux limites rencontrées dans les protocoles de routage classiques quand on passe à l'échelle. Le premier schéma de routage utilise une portée de transmission fixe alors que le deuxième ajuste la portée de transmission quand les nœuds qui sont proches de la station de base cessent de fonctionner.

Les résultats obtenus ont été très prometteurs comparativement à d'autres protocoles de routages tels que LEACH et LEACH-C.

Conclusion générale

L'objectif de ce mémoire portait sur la problématique du routage dans les réseaux de capteurs à grandes échelles. Pour y remédier nous avons proposé une démarche de routage basé sur les algorithmes génétiques sans la moindre hiérarchisation, qui respecte les caractéristiques principales de ce type de réseau et la contrainte énergétique en particulier.

L'implémentation de cet algorithme a montré son efficacité, où il garantit l'acheminement correct des données avec un taux de perte négligeable dans la plupart des scénarios testés tout en préservant la durée de vie du réseau pour des périodes assez importantes et qui surpasse même certains protocoles tel que LEACH.

La réalisation de ce projet de fin d'étude nous a été très bénéfique où on a pu maîtriser un nouveau simulateur, avoir de nouvelles connaissances sur le monde des réseaux et sur les réseaux de capteurs en particulier à travers des articles et des travaux réalisés auparavant.

Le thème traité est un thème de recherche qui nécessite une continuité et pourquoi pas l'exploration de nouvelles méthodes et critères afin d'améliorer les résultats obtenus.

Perspectives :

- Implémenter une version de l'algorithme qui favorise les nœuds qui se situent dans la même voie angulaire que la station de base.
- Essayer d'autres méthodes de croisements et de sélections.
- Explorer d'autres heuristiques.

Références bibliographiques

- [1] B. Rivollet et C. E. Serre, “Les réseaux de capteurs”, Rapport, Equipe CAPTIVEN, Juillet 2013.
- [2] http://gte.univ-littoral.fr/sections/documents-pdagogiques/chapitre-8-mesure/Les_capteurs.pdf, mise à jour 20/07/2013
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. “Wireless sensor networks: a survey”, *Computer Networks (Elsevier)*, vol.38, no.4, pp.393-422, March 2002.
- [4] K. Lin, J. Yu, J. Hsu, S. Zahedi, D. Lee, J. Friedman, A. Kansal, V. Raghunathan, and M. Srivastava, “Helimote : Enabling long-lived sensor networks through solar energy harvesting,” in *Proceedings of the 3rd ACM International Conference on Embedded Networked Sensor Systems (SenSys’05)*, pp.309–309, 2005.
- [5] Yacine Challal, http://moodle.utc.fr/file.php/498/SupportWeb/co/Module_RCSF_35.html, mise à jour 4/07/2012
- [6] C. Boano, J. Brown, Z. He, U. Roedig, and T. Voigt, “Low-Power radio communication in industrial outdoor deployments : The impact of weather conditions and ATEX-Compliance,” in *Sensor Applications, Experimentation, and Logistics*, ser. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Ed. Springer Berlin Heidelberg, vol. 29, pp.159–176, 2010.
- [7] C. Mettu. (2011), Telosb datasheet: <https://fr.scribd.com/doc/68138250/Telosb-Datasheet-t>
- [8] C.Y. Chong, S.P. Kumar. “Sensor Networks: Evolution, Opportunities, and Challenges”. *Proceedings of the IEEE*. 2003, Vol. 91, pp. 1247-1256.
- [9] T.B. Gosnell, J.M. Hall, C.L. Ham, D.A. Knapp, Z.M. Koenig, S.J. Luke, B.A. Pohl, A.Schach von Wittenau, and J.K. Wolford. “Gamma-Ray Identification of Nuclear Weapon Materials“. Lawrence Livermore National Lab. Livermore CA, USA., February, 1997. Technical report DE97053424.
- [10] Brown, M. J. “Users Guide Developed for the JBREWS Project”. Los Alamos National Laboratory of California University. 1999. Technical report LA-UR-99-4676.
- [11] P. Johnson and D.C Andrews. “Remote continuous monitoring in the home”, *Journal of Telemedicine and Telecare*, vol.2, no.2, pp.107-113, June 1996.
- [12] CORDIS : Service Projets et resultants, http://cordis.europa.eu/result/rcn/90958_fr.html, mise à jour 05/08/2015.
- [13] E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, and V.Z. Groza. “Sensor-based information appliances”, *IEEE Instrumentation Measurement Magazine*, vol.3, no.4, pp.31-35, December 2000.
- [14] Eya Dhib. “Routage avec QoS temps réel dans les réseaux de capteurs”, Mémoire d’ingénieur, option : “ingénierie de réseaux”, Ecole Supérieure des Communications de Tunis, 2007.

-
- [15] Akkaya K, Younis MA. “Survey on routing protocols for wireless sensor networks”. *Ad Hoc Networks* 2005;3(3):325–49
- [16] Adamu Murtala Zungeru, Li-Minn Ang, Kah Phooi Seng, “Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison”, *Journal of Network and Computer Applications*, Volume 35, Issue 5, September 2012, Pages 1508-1536
- [17] W. Heinzelman, J. Kulik, H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks”, in: *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’99)*, Seattle, WA, August 1999.
- [18] C. Intanagonwiwat, R. Govindan, D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks”, in: *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom_00)*, Boston, MA, August 2000.
- [19] D. Braginsky, D. Estrin, “Rumor routing algorithm for sensor networks, in: *Proceedings of the First Workshop on Sensor Networks and Applications (WSNA)*”, Atlanta, GA, October 2002.
- [20] C. Schurgers, M.B. Srivastava, “Energy efficient routing in wireless sensor networks”, in: *The MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force*, McLean, VA, 2001.
- [21] M. Chu, H. Haussecker, F. Zhao, “Scalable information driven sensor querying and routing for ad hoc heterogeneous sensor networks”, *The International Journal of High Performance Computing Applications* 16 (3) (2002) 293–313.
- [22] A. Manjeshwar, D.P. Agrawal, “TEEN: a protocol for enhanced efficiency in wireless sensor networks”, in: *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, April 2001.
- [23] Y. Yao, J. Gehrke, “The cougar approach to in-network query processing in sensor networks”, in: *SIGMOD Record*, September 2002.
- [24] R. Shah, J. Rabaey, “Energy aware routing for low energy ad hoc sensor networks”, in: *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, FL, March 2002.
- [25] N. Sadagopan et al., “The ACQUIRE mechanism for efficient querying in sensor networks”, in: *Proceedings of the First International Workshop on Sensor Network Protocol and Applications*, Anchorage, AK, May 2003.
- [26] D. Estrin et al., “Next century challenges: scalable coordination in sensor networks”, in: *Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom_99)*, Seattle, WA, August 1999.
- [27] B. Krishnamachari, D. Estrin, S. Wicker, “Modeling data centric routing in wireless sensor networks”, in: *Proceedings of IEEE INFOCOM*, New York, June 2002.

- [28] A. Buczak, V. Jamalabad, "Self-organization of a heterogeneous sensor network by genetic algorithms", in: C.H. Dagli et al. (Eds.), *Intelligent Engineering Systems Through Artificial Neural Networks*, vol. 8, ASME Press, New York, 1998, pp. 259–264.
- [29] C.R. Lin, M. Gerla, "Adaptive clustering for mobile wireless networks", *IEEE Journal on Selected Areas in Communications* 15 (7) (1997) 1265–1275.
- [30] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks", in: *Proceeding of the Hawaii International Conference System Sciences*, Hawaii, January 2000.
- [31] S. Lindsey, C.S. Raghavendra, "PEGASIS: power efficient gathering in sensor information systems", in: *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, March 2002.
- [32] S. Lindsey, C.S. Raghavendra, K. Sivalingam, "Data gathering in sensor networks using the energy*delay metric", in: *Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, April 2001.
- [33] A. Manjeshwar, D.P. Agrawal, "APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks", in: *Proceedings of the 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile computing*, Ft. Lauderdale, FL, April 2002.
- [34] L. Subramanian, R.H. Katz, "An architecture for building self configurable systems", in: *Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing*, Boston, MA, August 2000.
- [35] M. Younis, M. Youssef, K. Arisha, "Energy-aware routing in cluster-based sensor networks", in: *Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002)*, Fort Worth, TX, October 2002.
- [36] K. H. Chen, J. M. Huang and C. C. Hsiao, "CHIRON: An energy-efficient chain-based hierarchical routing protocol in wireless sensor networks", *Wireless Telecommunications Symposium, 2009. WTS 2009*, Prague, 2009, pp. 1-5.
- [37] S. Mao, and Y. T. Hou, "BeamStar: An Edge-Based Approach to Routing in Wireless Sensor Networks", *IEEE Transactions on Mobile Computing*, Vol. 6, Issue 11, pp. 1284-1296, 2007.
- [38] Hwang, So-Young; Jin, Gwang-Ja; Shin, Chang-Sub; Kim, Bong-Soo; "Energy-aware Tree Routing Protocol for Wireless Sensor Networks", *The Journal of Korean Institute of Communications and Information Sciences*, Vol. 33, no. 8B, pp.624-629, 2008.
- [39] Roychowdhury S, Patra C. "Geographic adaptive fidelity and geographic energy aware routing in ad hoc routing", In: *Special issue of IJCTT 2010*, for international conference (ACCTA'10), vol. 1, p. 3–5, 2010.

-
- [40] Y. Xu, J. Heidemann, D. Estrin, “Geography-informed energy conservation for ad hoc routing”, in: Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’01), Rome, Italy, July 2001;
- [41] Y. Yu, D. Estrin, R. Govindan, “Geographical and energyaware routing: a recursive data dissemination protocol for wireless sensor networks”, UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, May 2001.
- [42] Karp B, HT Kung. “GPSR: greedy perimeter stateless routing for wireless sensor networks”. In: Proceedings of the 6th annual ACM/IEEE international conference on mobile computing and networking (MobiCom’00). Boston (MA); 2000.
- [43] Dorigo M. “Solve difficult optimization problems. Lecture notes in computer science”, In Proceedings of the 6th European conference on advances in artificial life table of contents, vol. 2159; 2001. p. 11–22.
- [44] Guo W, Zhang W, Lu GA. “Comprehensive routing protocol in wireless sensor network based on ant colony algorithm”, In Proceedings of NSWCTC’10, Proceedings of the 2010 second international conference on networks security, wireless communications and trusted computing; 2010.
- [45] Wang L, Zhang R. Model A N. “An energy-balanced ant-based routing protocol for wireless sensor networks”. Network 2009; pp1–4.
- [46] Paone M, Paladina L, Scarpa M, Puliafito A. “A multi-sink swarm-based routing protocol for wireless sensor networks”. In: Proceedings of IEEE symposium on computers and communications; 2009. p. 28–33.
- [47] De-min G, Huan-yan Q, Xiao-yong Y, Xiao-nan W. “Based on ant colony multicast trees of wireless sensor network routing research”. Computer 2008.
- [48] Dorigo M., Maniezzo V. and A. Coloni, “*Positive feedback as a search strategy*”, Technical Report No 91-016, Politecnico di Milano, Italy, 1991.
- [49] Andrea Costanzo, Thé Van Luong, Guillaume Marill, “Optimisation par colonies de fourmis“, Mai 2006.
- [50] Sébastien Verel, Manuel Clergue, “Cours : Métaheuristiques pour l’optimisation combinatoire”, Département des Mathématiques et d’Informatique, Université les Antilles.
- [51] Mohammed Lehsaini, “Technique d’optimisation basées sur les heuristiques”, Cours M1 RSD, Département d’Informatique, Université de Tlemcen.
- [52] Marek Obitko, Introduction to Genetic Algorithms , <http://www.obitko.com/tutorials/genetic-algorithms/>, mise à jour Mai 2016.
- [53] Mohammed Lehsaini, “Les algorithmes génétiques”, Cours M1 RSD, Département d’Informatique, Université de Tlemcen
- [54] Andras Varga, OMNeT++ - Simulation Manual, <https://omnetpp.org/doc/omnetpp/manual/>, Mise à jour Avril 2016,

- [55] Wendi B. Heinzelman, *Member, IEEE*, Anantha P. Chandrakasan, *Senior Member, IEEE*, and Hari Balakrishnan, *Member, IEEE*, An Application-Specific Protocol Architecture For Wireless Microsensor Networks, October 2012.

Résumé :

Récemment les recherches dans le domaine du sans-fil ont abouti à une large évolution, ce qui a permis de concevoir de nouveaux types de réseaux tel que les réseaux de capteurs. Dans ce projet de fin d'étude nous présentons deux démarches de routage basées sur les algorithmes génétiques avec différentes approches dans le choix des chemins menant vers la station de base. La première est basée sur le critère énergie lors de l'établissement des routes. Alors que la deuxième repose sur le critère distance pour son choix de chemins. Nous montrons par suite les performances des algorithmes proposés et leurs efficacités en terme de durée de vie du réseau en se basant sur les résultats obtenus par la simulation sous OMNET++ des démarches proposées dans des topologies différentes. Enfin nous dressons une comparaison avec d'autres approches tel que LEACH et nous proposons quelques améliorations qui pourront être effectués afin d'accroître le rendement des capteurs en terme de consommation d'énergie.

Mots clés : sans-fil, capteurs, station de base, routage, algorithmes génétiques, énergie, distance, durée de vie

Abstract:

Recently wireless technology has been the topic of many researches which has contributed to appearance of wireless sensors network. In this work we will expose two different routing ways based on genetic algorithm. The first method promote energy criterion in the choice of the best path to the base station. The second is based on distance factor. After that we will prove the performance of our proposed algorithms in terms of network life time through simulation results. Finally we will compare our routing algorithms with other routing protocols such as LEACH and suggest some improvements which can conserve in better way sensors energies.

Key words: wireless, sensors, base station, routing, genetic algorithm, energy, distance, life time.

المخلص :

لقد شهد مجال اللاسلكي تطورا ملحوظا في السنوات الأخيرة مما أدى الى ظهور العديد من التحديات و الابتكارات في شبكات الاتصال نذكر منها شبكة الاستشعار اللاسلكية. في هذه المذكرة و بالاعتماد على الخوارزميات الجينية سنستعرض توجيهين مختلفين لإيصال المعلومات المستشعرة، الى قاعدة جمع المعلومات. الأول يركز بشكل أكبر على معيار الطاقة في حين أن الثاني يعتمد على قصر المسافة في اختيار الطرق المؤدية إلى القاعدة. سنقوم بعدها بإثبات فعالية الحلول المقترحة استنادا على النتائج المحصل عليها من خلال المحاكاة المجرى على أرضيات اختبار مختلفة ثم ننهي بإجراء مقارنة مع بروتوكولات أخرى بالإضافة الى اقتراح بعض التحسينات التي تمدد مدة حياة الشبكة وتمكن أجهزة الاستشعار من الاحتفاظ بقدر أكبر من الطاقة

الكلمات المفتاحية : اللاسلكي، الخوارزميات الجينية، مدة حياة الشبكة، أجهزة الاستشعار، قاعدة جمع المعلومات، الطاقة، المسافة