

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

Université A/Mira de Béjaia  
Faculté des Sciences Exactes  
Département d'Informatique

## *Mémoire de fin de cycle*

En vue de l'obtention du diplôme MASTER  
en Informatique

**Option :**

Systèmes Complexes et Technologies de l'Information et du Contrôle

## Thème

---

**Un protocole de routage à basse consommation d'énergie  
pour les réseaux de capteurs sans fil**

---

**Présenté par :**

*M<sup>r</sup> ALLAG Sofiane*

*M<sup>r</sup> MANSOUR Abdelaziz*

**Soutenu le :** 30/06/2012

**Devant le jury composé de :**

<b>Président :</b>	<i>M<sup>r</sup> MIR Foudil</i>	M.A.A, Université A. MIRA de Bejaia
<b>Rapporteur :</b>	<i>D<sup>r</sup> OMAR Mawloud</i>	M.C.B, Université A. MIRA de Bejaia
<b>Examineurs :</b>	<i>M<sup>r</sup> ABBACHE Bournane</i>	M.A.B, Université A. MIRA de Bejaia
	<i>M<sup>elle</sup> TIAB Amel</i>	Doctorante, Université A. MIRA de Bejaia

Année universitaire 2011/2012



Ah ! Les remerciements ! Ça permet à chaque étudiant de remémorer les moments importants qui ont ponctué ses années d'étude, ainsi que les personnes qui l'ont accompagné et encouragé. Ces années d'étude ont été l'occasion pour nous de rencontrer de nombreuses personnes qui nous ont beaucoup apporté du point de vue scientifique, mais également de partager des amitiés fortes et inoubliables.

Tout d'abord, nous remercions Dieu le tout puissant pour nous avoir donné la force et l'intelligence nécessaire à ce modeste travail.

Nous tenons à remercier très chaleureusement les membres du jury. Merci à Monsieur Mir Foudil qui nous a fait l'honneur de présider ce jury. Nos pensées vont également à Abbache Bournane et à mademoiselle Tiab Amel qui nous ont fait l'honneur de participer à ce jury et pour l'intérêt qu'ils ont porté à ce mémoire.

Nous tenons à dire toute nos gratitudes à notre promoteur Monsieur Omar Mawloud. Merci ! Pour nous avoir encadré au sens propre du terme, puis pour nous avoir fait confiance et pour nous avoir accompagné, encouragé et conseillé au cours de cette année tant sur le plan technique que humain.

Le docteur Omar Mawloud a su guider notre travail avec intelligence et a fait preuve de disponibilité en toute circonstance. Sinon, que mettre en avant parmi ses qualités ? Son enthousiasme communicatif, son énergie débordante, ses qualités humaines.

Nous tenons aussi à remercier tous nos professeurs et toute l'équipe pédagogique de l'université qui ont travaillé avec abnégation pour nous permettre de suivre ce cursus.

De même, nous remercions ceux qui ont contribué à ce travail de près ou de loin, soit avec leur support, leur amitié ou leur amour.

*Merci.*



*Je dédie ce modeste travail :*

*À la mémoire de ma mère,*

*À mon père,*

*À mes frères et sœurs,*

*À toute la famille,*

*À mes amis et collègues, et tous ceux qui m'ont aidé,*

*À mon binôme Abdelaziz et sa famille.*

*ALLAG Sofiane*



*Je dédie ce modeste travail :*

*À mes parents,*

*À mes frères et sœurs,*

*À toute la famille,*

*À mes amis et collègues, et tous ceux qui m'ont aidé,*

*À mon binôme Sofiane et sa famille.*

*MANSOUR Abdelaziz*

## Résumé

Un réseau de capteurs sans fil est un réseau constitué de nœuds capteurs déployés en grand nombre dans le but de collecter et de transmettre des données vers une destination. Les nœuds capteurs composants le réseau possèdent généralement de faibles capacités de calcul, de mémoire et d'énergie. Dans ce type de réseau, l'énergie est la ressource la plus critique. La défaillance d'un simple capteur, à cause de l'épuisement de sa batterie, peut probablement faire partitionner une partie du réseau du reste, par conséquent, son utilisation doit être optimisée pour maximiser la durée de vie du réseau. L'objectif de ce mémoire est d'étudier les contraintes énergétiques liées à l'utilisation des batteries à capacité limitée pour l'alimentation des capteurs. Ainsi dans notre travail, nous avons proposé un nouveau protocole de routage à basse consommation d'énergie où le capteur est allégé de toute décision concernant le routage, c'est à la station de base de choisir les chemins et elle transmet juste une valeur entière (pas tout le chemin) issue de l'utilisation d'une fonction spécifique sur les identifiants du chemin sélectionné, ensuite le nœud qui reçoit cette valeur déduira le prochain nœud successeur ainsi que la nouvelle valeur entière à transmettre à ce dernier. Chaque nœud procédera de la même façon jusqu'à ce que le paquet atteigne la destination.

**Mots clés :** Réseaux de capteurs sans fil, protocole de routage, consommation d'énergie.

## abstract

Wireless sensor networks is a network consisting of nodes sensors deployed in great number in order to collect and transmit data to a destination. The nodes sensors components the network generally have low computing capacity, memory and energy. In this type of network, the energy is the most critical resource. The failure of a single sensor, due to the depletion of its battery, can probably make a partition of the network. Therefore, its use must be optimized to maximize the lifetime of the network. The aim of this report is to focus on energy constraints related to the use of batteries with limited capacity for the supply of sensors. In our work, we have proposed a new routing protocol with low consumption energy where the sensor is relieved of any decision concerning the routing, it is at the basic station to choose the paths and transmit just a single value (not all the path) resulting from the use of a specific function on the identifiers of the selected path, then the node which receive this integer value will deduce the next node successor, and the new integer value to transmit to the latter. Each node will proceed the same way until the packet reaches the destination.

**Keywords :** Wireless sensor network, routing protocol, energy consumption.

# Table des matières

<b>Table des matières</b>	
<b>Liste des tableaux</b>	<b>ii</b>
<b>Table des figures</b>	<b>iii</b>
<b>Liste des abréviations</b>	<b>iv</b>
<b>Introduction Générale</b>	<b>1</b>
<b>I Généralités sur les réseaux de capteurs sans fil</b>	<b>3</b>
I.1 Introduction . . . . .	3
I.2 Les capteurs . . . . .	3
I.2.1 Définition . . . . .	3
I.2.2 Les composants d'un capteur . . . . .	3
I.3 Les réseaux de capteurs sans fil . . . . .	4
I.3.1 Définition . . . . .	4
I.3.2 Architecture . . . . .	5
I.3.3 Types d'application des RCSFs . . . . .	5
I.3.4 Spécificités des RCSFs . . . . .	6
I.3.5 La consommation d'énergie dans les RCSFs . . . . .	6
I.3.6 Quelques domaines d'applications des RCSFs . . . . .	7
I.3.7 Facteurs de conception des réseaux de capteurs sans fil . . . . .	8
I.3.8 Quelques problématiques dans les RCSFs . . . . .	9
I.4 Conclusion . . . . .	10
<b>II Etat de l'art sur le routage dans les RCSFs</b>	<b>11</b>
II.1 Introduction . . . . .	11
II.2 La classification des protocoles de routage dans les RCSFs . . . . .	11
II.2.1 Classification selon la structure du réseau . . . . .	12
II.2.1.1 Routage à plat . . . . .	12
II.2.1.2 Routage hiérarchique . . . . .	12
II.2.1.3 Routage basé sur la localisation . . . . .	13
II.2.2 Classification selon les fonctions du protocole . . . . .	13
II.2.2.1 Routage basé sur la Qualité de Service . . . . .	13
II.2.2.2 Routage basé sur le flux de données dans le réseau . . . . .	13

II.2.2.3	Routage multi-chemin . . . . .	13
II.2.2.4	Routage basé sur la négociation . . . . .	14
II.2.3	Classification selon l'établissement de la route . . . . .	14
II.2.3.1	Protocoles proactifs . . . . .	14
II.2.3.2	Protocoles réactifs . . . . .	14
II.2.3.3	Protocoles hybrides . . . . .	14
II.2.4	Classification selon l'initiateur des communications . . . . .	14
II.2.4.1	Communication lancée par la source . . . . .	14
II.2.4.2	Communication lancée par la destination . . . . .	15
II.3	Quelques protocoles de routage pour les RCSFs . . . . .	15
II.3.1	Inondation(Flooding) . . . . .	15
II.3.2	Gossiping . . . . .	16
II.3.3	Diffusion dirigée (Direct Diffusion) . . . . .	16
II.3.4	SPIN : <i>Sensor Protocols for Information via Negotiation</i> . . . . .	17
II.3.5	Routage par rumeur . . . . .	18
II.3.6	GEAR : <i>Geographic and Energy-Aware Routing</i> . . . . .	19
II.3.7	LEACH : <i>Low Energy Adaptive Clustering Hierarchy</i> . . . . .	19
II.3.8	PEGASIS : <i>Power-Efficient Gathering in Sensor Information System</i> . . . . .	20
II.3.9	TEEN : <i>Threshold sensitive Energy Efficient sensor Network protocol</i> et APTEEN : <i>Adaptive Periodic Threshold-sensitive Energy Efficient</i> <i>Sensor Network Protocol</i> . . . . .	21
II.3.10	SAR : <i>Sequential Assignment Routing</i> . . . . .	21
II.3.11	SPEED : <i>A stateless protocol for real-time communication in sensor</i> <i>networks</i> . . . . .	22
II.4	Synthèse des protocoles de routage dans les RCSFs . . . . .	22
II.4.1	Classification . . . . .	22
II.4.2	Comparaison . . . . .	23
II.5	Conclusion . . . . .	24
<b>III</b>	<b>Protocole de routage pour l'économie d'énergie dans les RCSFs</b>	<b>25</b>
III.1	Introduction . . . . .	25
III.2	Problématique . . . . .	25
III.3	Motivations . . . . .	26
III.4	Estimation de distances avec RSSI . . . . .	26
III.5	Modèle énergétique . . . . .	26
III.6	Hypothèses . . . . .	27
III.7	Description du protocole . . . . .	28
III.7.1	Phase d'installation . . . . .	29
III.7.2	Phase de transmission . . . . .	38
III.7.2.1	Etape 1 : Choix du chemin optimal . . . . .	38
III.7.2.2	Etape 2 : Routage de paquet . . . . .	40
III.7.3	Phase de maintenance . . . . .	43
III.8	Conclusion . . . . .	44
	<b>Conclusion générale et perspectives</b>	<b>44</b>

**Bibliographie**

**46**



# Liste des tableaux

II.1	Classification des différents protocoles de routage dans les RCSFs . . . . .	23
II.2	Comparaison entre les différents protocoles de routage dans les RCSFs . . . . .	24

# Table des figures

I.1	Les composants d'un nœud capteur. . . . .	4
I.2	Architecture d'un réseau de capteurs sans fil. . . . .	5
II.1	Classification des approches de routage dans les RCSFs. . . . .	12
II.2	Problèmes de routage par inondation. . . . .	15
II.3	Les phases du routage de la Diffusion Dirigée . . . . .	17
II.4	Un exemple d'exécution du protocole SPIN. . . . .	18
II.5	Un exemple de routage par rumeur. . . . .	19
III.1	Modèle de consommation d'énergie. . . . .	27
III.2	Un exemple illustratif de la diffusion du paquet <i>INST</i> . . . . .	31
III.3	La matrice de topologie . . . . .	33
III.4	Un exemple illustratif de la diffusion du paquet <i>INST</i> . . . . .	35
III.5	Exemple illustratif de l'envoi du paquet <i>repINST</i> . . . . .	36
III.6	Exemple de la minimisation des tailles des tables de voisinages. . . . .	37
III.7	La construction de la matrice de topologie. . . . .	38
III.8	Exemple d'application de la procédure Max (Min( )). . . . .	40
III.9	Exemple de l'acheminement des paquets . . . . .	41
III.10	Exemple d'acheminement des paquets(deuxième solution) . . . . .	43

# Liste des abréviations

<b>ADC</b>	<b>A</b> nalog- <b>D</b> igital <b>C</b> onverter
<b>ADV</b>	<b>A</b> dvertising
<b>APTEEN</b>	<b>A</b> daptive <b>P</b> eriodic <b>T</b> hreshold-sensitive <b>E</b> nergy <b>E</b> fficient sensor <b>N</b> etwork protocol
<b>DD</b>	<b>D</b> irected <b>D</b> iffusion
<b>GAF</b>	<b>G</b> eographic <b>A</b> daptative <b>F</b> idelity
<b>GEAR</b>	<b>G</b> eographic and <b>E</b> nergy <b>A</b> ware <b>R</b> outing
<b>GPS</b>	<b>G</b> lobal <b>P</b> osition <b>S</b> ystem
<b>GSM</b>	<b>G</b> lobal <b>S</b> ystem of <b>M</b> obile communications
<b>ht</b>	<b>h</b> ard <b>t</b> hreshold
<b>INST</b>	<b>I</b> nstallation
<b>LEACH</b>	<b>L</b> ow <b>E</b> nergy <b>A</b> daptive <b>C</b> lustering <b>H</b> ierarchy
<b>MAC</b>	<b>M</b> edia <b>A</b> ccess <b>C</b> ontrol
<b>MAX</b>	<b>M</b> aximum
<b>MIN</b>	<b>M</b> inimum
<b>PEGASIS</b>	<b>P</b> ower- <b>E</b> fficient <b>G</b> Athering in <b>S</b> ensor <b>I</b> nformation <b>S</b> ystem
<b>SPIN</b>	<b>S</b> ensor <b>P</b> rotocols for <b>I</b> nformation via <b>N</b> egotiation
<b>QoS</b>	<b>Q</b> ualité <b>d</b> e <b>S</b> ervice
<b>RCSFs</b>	<b>R</b> éseaux de <b>C</b> apteurs sans <b>F</b> il
<b>rep</b>	<b>r</b> eponse
<b>REQ</b>	<b>R</b> equête
<b>req</b>	<b>r</b> equête
<b>RSSI</b>	<b>R</b> eceived <b>S</b> ignal <b>S</b> trength <b>I</b> ndication
<b>SAR</b>	<b>S</b> equential <b>A</b> ssignement <b>R</b> outing
<b>SB</b>	<b>S</b> tation de <b>B</b> ase
<b>SNR</b>	<b>S</b> ignal to <b>N</b> oise <b>R</b> ation
<b>st</b>	<b>s</b> oft <b>t</b> hreshold
<b>TEEN</b>	<b>T</b> hreshold-sensitive <b>E</b> nergy- <b>E</b> fficient sensor <b>N</b> etwork
<b>WiFi</b>	<b>W</b> ireless <b>F</b> idelity
<b>WSN</b>	<b>W</b> ireless <b>S</b> ensor <b>N</b> etworks

# Introduction Générale

Au cours de ces dernières années, les technologies des réseaux sans fils n'ont jamais cessé de croître grâce aux développements technologiques dans divers domaines liés à la micro-électronique. En plus, avec l'émergence des réseaux de capteurs sans fils (RCSFs ou WSN : Wireless Sensor Networks), de nouvelles thématiques ont été ouvertes et de nouveaux défis ont vu le jour pour répondre aux besoins des personnes et aux exigences de plusieurs domaines d'application (industriel, culturel, environnemental) tel que : le contrôle d'intrusions, le calcul de températures, le calcul de changements climatiques, la surveillance des déplacements d'animaux (avec récepteurs GPS), surveillance de malades, etc.

Cependant, la maîtrise de la consommation d'énergie par les réseaux capteurs et la maximisation de leur durée de vie restent les problématiques les plus fondamentales car les capteurs sont de petits composants avec une faible capacité de stockage, de calcul et sont alimentés par des batteries dont la capacité est très limitée et qui sont généralement difficile voire impossible de les changer. En plus ils sont conçus pour être déployés d'une manière dense dans des endroits hostiles et difficiles à l'accès d'où la nécessité de limiter au maximum leurs dimensions physiques qui sont déterminées au détriment des capacités de calcul, de mémoire et de ressources énergétiques. Donc, pour qu'un réseau de capteurs reste autonome pendant une longue durée (quelques mois ou quelques années) et ait par la suite une longévité maximale, il faut que la consommation d'énergie soit prise en compte à tous les niveaux de l'architecture réseaux (de la couche physique à la couche application). De se fait, plusieurs facteurs interviennent dans ces différent niveaux tel que l'état du module radio, le type d'application, la politique d'accès au canal de transmission, le protocole de routage, etc.

Notre travail entre dans le cadre de l'étude du problème de routage dans les réseaux de capteurs sans fils. Notre étude offre principalement, une étude synthétique des travaux de recherche qu'ont été fait, de proposer un nouveau protocole de routage à basse consommation d'énergie, dans le but de résoudre le problème d'acheminement de données entre les nœuds du réseau avec la prise en compte de la contrainte énergétique.

Notre mémoire est structuré comme suit : le premier chapitre présente une introduction au domaine des réseaux de capteurs sans fils. Nous commençons d'abord par la définition des différentes notions et concepts gravitant autour de cette thématique, ensuite nous exposons quelques domaines d'application des RCSFs, les facteurs pris lors de la conception de ces derniers et les problématiques soulevées dans ce type de réseau.

Le deuxième chapitre décrit dans le premier volet les différentes classes des protocoles de routage avec une description plus au moins détaillée. Dans le second volet, ce chapitre passe en revue quelques protocoles de routage dans les RCSFs qui permettent la minimisation de la consommation d'énergie. Enfin, une classification et une comparaison entre ces derniers est présentée.

Dans le troisième chapitre, nous présentons notre protocole de routage sous contrainte énergétique dans les RCSFs avec une description détaillée de ces différentes phases.

# Chapitre I

## Généralités sur les réseaux de capteurs sans fil

### I.1 Introduction

Les réseaux de capteurs sans fil sont parmi les technologies les plus importantes de nos jours. Ces réseaux sont constitués d'un grand nombre de nœuds qui sont des micro-capteurs capable de récolter et de transmettre des données de manière autonome. Les nœuds capteurs composants le réseau possèdent généralement de faibles capacités de calcul, de mémoire et d'énergie. Ces contraintes matérielles ont influencé une grande partie des problématiques de recherche du domaine tel que le routage.

Le but de ce chapitre est de présenter cette nouvelle technologie qui est susceptible d'être déployée dans de divers domaines applicatifs. Pour mieux cerner les enjeux du sujet, nous présenterons dans ce qui suit, ce que sont les capteurs, les réseaux de capteurs, ainsi qu'un aperçu sur l'ensemble de notions nécessaires à la compréhension de la suite du mémoire.

### I.2 Les capteurs

#### I.2.1 Définition

Un capteur sans fil est un petit dispositif électronique capable de mesurer une valeur physique environnementale et de la communiquer à un centre de contrôle via une station de base [29]. Ils sont capables de surveiller une grande variété de phénomènes ambiants, notamment : la température, l'humidité, le mouvement des véhicules, la pression, le taux de bruits, la présence ou absence de certains types d'objets, et d'autres caractéristiques, tel que la vitesse, la direction et le volume d'un objet donné.

#### I.2.2 Les composants d'un capteur

Un capteur est composé de quatre unités de base (figure I.1 [29]) :

- **L'unité d'acquisition** : elle est composée d'un capteur qui obtient des mesures sur les paramètres environnementaux et d'un convertisseur Analogique/Numérique ADCs (Analog-Digital Converter) qui convertit l'information relevée et la transmet à l'unité de traitement.
- **L'unité de traitement** : elle est chargée d'exécuter les protocoles de communications qui permettent de faire collaborer le capteur avec d'autres capteurs. Elle peut aussi analyser les données captées.
- **Un module de communication (Transceiver)** : il est responsable de toutes les émissions et réceptions de données via un support de communication radio.
- **Batterie** : elle alimente les unités que nous avons citées et elle n'est généralement ni rechargeable ni remplaçable.

Il existe des capteurs qui sont dotés d'autres composants additionnels tels que les systèmes de localisation GPS (Global Position System).

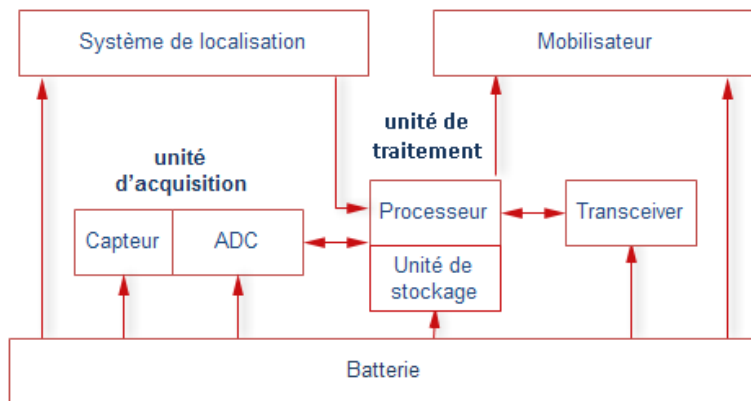


FIGURE I.1 – Les composants d'un nœud capteur.

## I.3 Les réseaux de capteurs sans fil

### I.3.1 Définition

Un réseau de capteurs se définit comme un ensemble de capteurs connectés entre eux, où chaque capteur étant muni d'un émetteur-récepteur [10]. Les réseaux de capteurs sans fil ou "*Wireless Sensor Network*" sont considérés comme un type spécial des réseaux Ad hoc où l'infrastructure fixe de communication et l'administration centralisée sont absentes et les nœuds jouent, à la fois, le rôle des hôtes et des routeurs [17]. Chaque nœud est capable de surveiller son environnement et de réagir en cas de besoin en envoyant l'information collectée à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil [29].

### I.3.2 Architecture

La figure I.2 [9] représente une architecture typique d'un réseau de capteurs sans fil. Ce réseau communique à travers une passerelle avec un réseau d'un autre type qui mène vers une unité contrôlée par l'utilisateur. Le réseau intermédiaire qui sert à déporter le poste utilisateur du lieu de déploiement du réseau de capteurs peut être le réseau GSM, un réseau sans fil comme le WiFi, le réseau internet...etc.

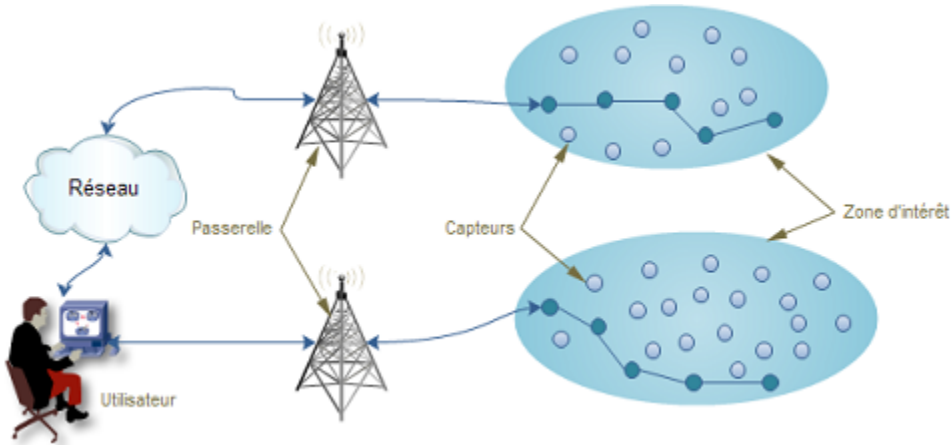


FIGURE I.2 – Architecture d'un réseau de capteurs sans fil.

### I.3.3 Types d'application des RCSFs

La méthode de captage des données dans un RCSF dépend de l'application et de l'importance de la donnée. De ce fait, nous distinguons quatre classes d'applications : orientées temps (*time driven*), orientées événements (*event driven*), orientées requêtes (*query driven*) et hybrides [19].

- **Applications orientées temps** : cette classe représente les applications où l'acquisition et la transmission des données capturées sont liées au temps : instant précis, période d'acquisition. Cette période d'acquisition peut être plus au moins longue selon l'application (de quelques secondes jusqu'à quelques heures voire des jours).
- **Applications orientées événements** : dans ce type d'applications, les capteurs envoient leurs données seulement si un événement spécifique se produit. Nous pouvons citer l'exemple de surveillance des feux dans les forêts où un capteur envoie des alarmes à la station de base dès que la température dépasse un certain seuil.
- **Applications orientées requêtes** : dans ce cas, un capteur envoie de l'information uniquement suite à une demande explicite de la station de base. Cette classe d'application est destinée aux applications adaptées à l'utilisateur.
- **Applications hybrides** : ce type d'application met en œuvre les trois types d'applications décrits précédemment. Par exemple, dans un réseau conçu pour le suivi d'objets,



le réseau peut combiner entre un réseau de surveillance (*time driven*) et un réseau de collecte de données par événements (*event driven*). Par exemple, pendant les longues périodes d'inactivité des capteurs et lorsque aucun objet n'est présent, le réseau peut assurer une fonction de surveillance.

### I.3.4 Spécificités des RCSFs

Les principales caractéristiques des réseaux de capteurs sont :

- **Densité « importante » des nœuds** : le nombre de capteurs utilisés dans les réseaux de capteurs sans fil peut varier de quelques entités à plusieurs dizaines de milliers [21].
- **Topologie dynamique** : la mobilité des nœuds, la possibilité d'étendre le réseau par ajout de nouveaux nœuds et la suppression de nœuds défaillants, changent fréquemment la topologie des réseaux. Ceci nécessite une maintenance permanente pour assurer le bon fonctionnement des réseaux de capteurs.
- **Une énergie limitée** : les capteurs sont équipés de batteries avec une énergie limitée (plusieurs jours à quelques années). De plus, les réseaux de capteurs quand ils sont déployés, le sont souvent dans des zones difficiles d'accès pour l'homme. Il est donc difficile de pouvoir changer les batteries des capteurs[21].
- **Faible puissance de calcul** : malgré les progrès récents dans la fabrication de capteurs de plus en plus puissant, les capteurs actuels souffrent d'un manque de puissance de calcul (par exemple seulement 16 Mhz de puissance et 128 Koctets de mémoire programmable pour un capteur MicaZ<sup>1</sup>). Cette faible puissance ne permet pas d'utiliser des algorithmes complexes[21].

### I.3.5 La consommation d'énergie dans les RCSFs

L'énergie consommée par un capteur est principalement due aux opérations suivantes : le captage, le traitement et la communication de données[29].

- **Energie de capture** : l'énergie consommée au moment du captage varie suivant la nature de l'application. Un captage sporadique consomme moins d'énergie qu'un contrôle d'événement constant. La complexité de l'événement à détecter joue également un rôle crucial pour déterminer la quantité d'énergie consommée.

Les environnements contenant un niveau de bruit élevé entraîne l'augmentation de l'énergie nécessaire pour cette phase [17].

- **Energie de traitement** : l'énergie de traitement est composée de deux sortes d'énergie : l'énergie de commutation et l'énergie de fuite. L'énergie de commutation est déterminée par la tension d'alimentation et la capacité totale commutée au niveau logiciel (en exécutant un logiciel). Par contre, l'énergie de fuite correspond à l'énergie consommée lorsque l'unité de calcul n'effectue aucun traitement [29].

---

<sup>1</sup>Un capteur de Crossbow, de taille  $\sim 10\text{cm}^3$  avec une mémoire vive de 4kB et un processeur de 4MHz.[15]

- **Energie de communication** : l'énergie de communication se décline en trois parties : l'énergie de réception, l'énergie de l'émission et l'énergie en état de veille. Cette énergie est déterminée par la quantité des données à communiquer et la distance de transmission, ainsi que par les propriétés physiques du module radio. L'émission d'un signal est caractérisée par sa puissance ; quand la puissance d'émission est élevée, le signal aura une grande portée et l'énergie consommée sera plus élevée [29].

Parmi les trois opérations citées auparavant, l'opération de communication de données est celle qui consomme la plus grande quantité d'énergie, ceci, à cause de la multitude de composants électroniques intégrés au circuit responsable de cette action.

### I.3.6 Quelques domaines d'applications des RCSFs

La taille de plus en plus réduite des micros-capteurs, leur coût de plus en plus faible, la large gamme des types de capteurs disponibles (thermiques, optiques, ...etc) ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs d'envahir rapidement plusieurs domaines d'applications [9]. Parmi les domaines où ces réseaux peuvent offrir les meilleurs contributions nous pouvons citer les domaines suivants :

- **Militaire** : le domaine militaire a été un moteur initial pour le développement des réseaux de capteurs comme c'est le cas pour plusieurs technologies [9]. Un exemple typique d'application des RCSFs dans ce domaine est le déploiement d'un tel réseau en un endroit stratégique ou difficile d'accès, afin de surveiller toutes les activités des forces ennemies en collectant des informations sur leurs positions et leurs mouvements, ou d'analyser le terrain avant d'y envoyer des troupes afin d'examiner la présence des agents chimiques, biologiques ou nucléaires.
- **Médical** : comme exemple d'application nous pouvons énumérer : les micro-caméras qui peuvent être ingérées, ayant la capacité sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain. La création d'une rétine artificielle composée d'une centaine de micro-capteurs pour améliorer la vision [9]. Le Suivi et la surveillance des médecins et des patients au sein de l'hôpital [17].
- **Environnemental** : dans ce domaine nous pouvons mentionner différents types d'applications comme : la dispersion de capteurs thermiques à partir d'un avion sur une forêt pour signaler un éventuel début d'incendie avant qu'il se propage et devienne incontrôlable. Le semis de nœuds capteurs en même temps que les graines dans les champs agricoles pour pouvoir identifier les zones sèches et rendre ainsi l'irrigation plus efficace. Le déploiement de nœuds capteurs sur les sites industriels, les centrales nucléaires ou dans les raffineries de pétrole afin de détecter des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.) et rapporter son origine à l'utilisateur dans un délai suffisamment court pour permettre une intervention efficace[9].
- **La sécurité** : parmi les exemples d'utilisation des réseaux de capteurs pour ce type d'applications, la détection des altérations dans la structure d'un bâtiment, suite à un séisme ou au vieillissement, par des capteurs intégrés dans les murs ou dans le béton.

- **Écologique** : un exemple de ce type d'applications est l'intégration de plusieurs micro-capteurs dans le système de climatisation et de chauffage des immeubles. Ainsi, la climatisation ou le chauffage ne sont déclenchés qu'aux endroits occupés par des personnes présentes et seulement si c'est nécessaire. Le système distribué peut aussi maintenir une température homogène dans les pièces. Utilisée à grande échelle, une telle application permettrait probablement de réduire la demande mondiale en énergie [9].
- **La localisation** : les réseaux de capteurs peuvent être très utiles dans le domaine de la localisation, ils peuvent par exemple de localiser les victimes enterrées sous la neige suite à une avalanche, en équipant les personnes susceptibles de se trouver dans des zones à risque par des capteurs. En facilitant ainsi, aux équipes de sauvetage la localisation des victimes[9].
- **Contrôle de l'habitat (maison intelligente)** : les RCSF sont appliqué en sein d'un habitat pour permettre par exemple : la gestion de l'éclairage, du chauffage, et les systèmes de refroidissement dans la maison. D'automatiser le contrôle de plusieurs systèmes destinés à améliorer la commodité et la sécurité. De surveiller une grande variété de conditions [16].

### I.3.7 Facteurs de conception des réseaux de capteurs sans fil

La conception et la réalisation des réseaux de capteurs sans fil sont influencées par plusieurs paramètres, parmi lesquels nous citons :

- **La tolérance aux fautes** : dans les RCSFs , un ou plusieurs capteurs peuvent ne pas fonctionner correctement. Dans ce cas de figure, le réseau doit être capable de détecter ce type d'erreur est d'y remédier, en cherchant par exemple à modifier ses tables de routages afin de maintenir le réseau toujours opérationnel.
- **La scalabilité** : la surveillance d'un phénomène peut nécessiter le déploiement d'un nombre important de nœuds qui est de l'ordre de plusieurs milliers de capteurs. Suivant l'application, ce nombre peut encore augmenter jusqu'à des millions de capteurs [17], les nouveaux schémas doivent pouvoir garantir un bon fonctionnement avec ce nombre de capteurs élevés.
- **La topologie du réseau** : la topologie des réseaux de capteurs peut changer au cours du temps pour les raisons suivantes : la défaillance d'un nœud capteur, la mobilité des nœuds capteurs ou l'épuisement de l'énergie d'un capteur.
- **Bande passante limitée** : afin de minimiser l'énergie consommée lors de transfert de données entre les nœuds, les capteurs opèrent à bas débit.
- **Ressources limitées** : en plus de l'énergie, les nœuds capteurs ont aussi une capacité de traitement et de mémoire limitée.

Ces facteurs importants servent comme directives pour le développement des algorithmes et protocoles utilisés dans les réseaux de capteurs, ils sont considérés également comme métriques de comparaison de performances entre les différents travaux dans le domaine [17].

### I.3.8 Quelques problématiques dans les RCSFs

Les recherches dans le domaine des réseaux de capteurs ont révélé plusieurs problématiques, parmi les problématiques cruciales, nous pouvons citer :

- **Routage** : le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau au sens d'un certain critère de performance. D'où les protocoles de routages conçus doivent, en plus de leurs fonctions classiques, participer à la synthèse et l'agrégation des données retournées aux utilisateurs, tout en considérant d'autres facteurs tels que les limitations matérielles, la mobilité et la consommation d'énergie. En assurant un routage performant en termes de minimisation de la consommation de l'énergie, du choix des routes optimales pour l'acheminement de l'information d'un capteur à la station de base et vice versa, de réduction du délai de délivrance des paquets...etc. Ainsi le réseau doit passer à l'échelle sans que ses performances se dégradent.
- **Couche MAC** : la spécificité des réseaux de capteurs sans fil nécessite le développement de nouveaux protocoles MAC qui s'adaptent aux contraintes imposées par ces réseaux. Ceci dans le but d'améliorer le débit, minimiser la consommation d'énergie, optimiser le partage du médium ainsi que minimiser le délai de délivrance des paquets[20].
- **Diffusion de l'information** : les protocoles de diffusion conçus pour les réseaux de capteurs doivent tenir compte de leurs spécificités ainsi que de leurs contraintes intrinsèques imposées [20].
- **Sécurité** : comme les nœuds sont dispersés dans une zone publique, ils doivent être capables de maintenir privées les informations qu'ils recueillent. Par conséquent, pour les applications qui exigent un niveau de sécurité assez élevé telles que les applications militaires, des mécanismes d'authentification, de confidentialité, et d'intégrité doivent être mis en place au sein de leur communauté. La puissance de calcul limitée des capteurs ouvre de véritables défis pour concevoir des algorithmes de cryptographie et des politiques de confiance spécifiques à ces réseaux.
- **La localisation** : une problématique liée à la localisation est celle du placement des nœuds dans un réseau de capteurs, c'est à dire comment positionner les nœuds les uns par rapport aux autres. Plusieurs applications dans les réseaux de capteurs dépendent lourdement sur l'habilité des nœuds à connaître leur position ou à découvrir la topologie globale du réseau. Au moment où ce problème peut être contourné par l'utilisation d'un système GPS dans les réseaux installés en plein air, cette solution ne peut être adaptée à ceux déployés à l'intérieur des bâtiments [17], car il cumule des handicaps, et il est disponible seulement en extérieur.
- **Synchronisation** : la synchronisation du temps est un challenge important et coûteux dans les réseaux de capteurs à communication multi-saut. De nombreuses applications de réseaux de capteurs demandent une synchronisation des horloges locales des nœuds. Par exemple, les nœuds capteurs collaborent entre eux pour parvenir à une tâche de détection plus complexe[2].

- **La couverture** : la problématique de la couverture consiste à profiter de la redondance, issue du déploiement aléatoire des nœuds sur la zone surveillée, pour procéder à leur mise en veille alternée. En d'autres termes, il s'agit d'ordonnancer les nœuds dans des ensembles d'activation disjoints, tout en respectant les contraintes de couverture et de connectivité [2].

Toutes ces problématiques sont des domaines de recherche actifs. Dans la suite, nous nous intéresserons particulièrement à celle de routage.

## I.4 Conclusion

Les réseaux de capteurs sans fil sont voués à se développer rapidement du fait de la grande variété des domaines d'applications. Dans ce chapitre, nous avons défini ces réseaux et leurs domaines d'application, nous avons cité leurs principales caractéristiques, ainsi que les différents facteurs de conception. Nous avons présenté les différentes problématiques révélées dans ce domaine, tel que le routage. Ce dernier est considéré parmi les problèmes posés à l'heure actuelle dans ce type de réseaux. Dans le chapitre suivant, nous présenterons les principaux protocoles de routage utilisés dans ces réseaux, ainsi que leurs classifications.

# Chapitre II

## Etat de l'art sur le routage dans les RCSFs

### II.1 Introduction

Le routage est une méthode d'acheminement des informations vers une destination donnée dans un réseau de connexion. Le développement des protocoles de routage spécifique aux réseaux de capteurs a attiré une grande part d'intention parmi les chercheurs dans le domaine. Mais les limites imposées par l'architecture matérielle des capteurs, en particulier celle de l'énergie et l'hostilité des environnements, figurent parmi les facteurs primordiaux à prendre en compte lors de la conception d'un protocole de routage. De ce fait, les protocoles élaborés doivent assurer une consommation minimale d'énergie tout en maintenant le bon fonctionnement du réseau et sans dégrader ses performances.

Ce chapitre présente un état de l'art sur les protocoles de routage qui minimisent la consommation d'énergie dans les RCSFs. Il fournit une classification sur la conception de ces protocoles en décrivant le principe général de quelques uns, ainsi que la comparaison entre eux par rapport à un ensemble de critères qui ont un impact considérable sur les performances de routage en termes de consommation d'énergie.

### II.2 La classification des protocoles de routage dans les RCSFs

Dans les réseaux de capteurs sans fil, chaque nœud communique directement avec son voisin, et pour communiquer avec d'autres nœuds, il lui est nécessaire de faire passer les données par d'autres nœuds qui se chargeront de les acheminer. Pour cela, il est primordial que les nœuds se situent les uns par rapport aux autres et soient capables de construire des liens entre eux, c'est le rôle d'un protocole de routage.

L'objectif d'un protocole de routage est de trouver des chemins qui mènent vers la destination et qui optimisent la métrique de consommation d'énergie. Plusieurs stratégies de routage

ont été proposées pour les réseaux de capteurs sans fil. Leurs principes de fonctionnement diffèrent suivant la philosophie de la classe à laquelle ils appartiennent. Ces approches peuvent être classifiées suivant la structure du réseau, les fonctions des protocoles, l'établissement des routes et l'initiateur des communications (Figure II.1 [23]).

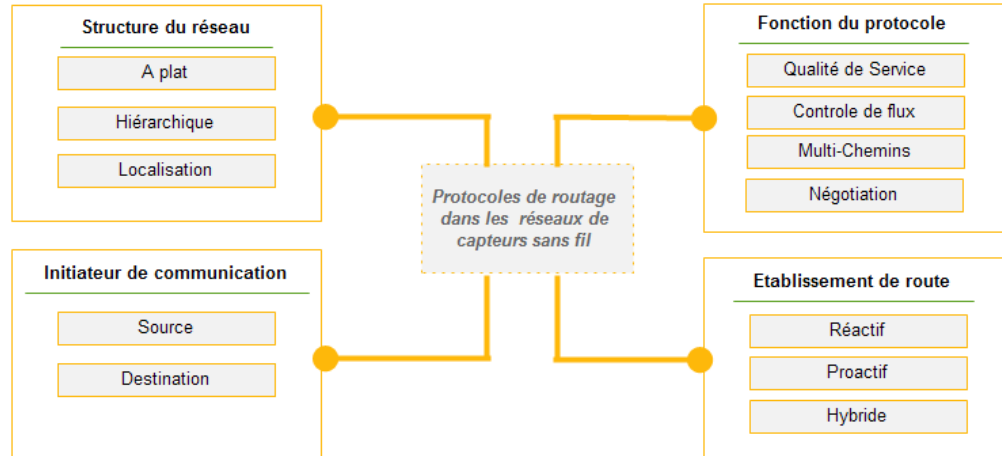


FIGURE II.1 – Classification des approches de routage dans les RCSFs.

## II.2.1 Classification selon la structure du réseau

Les protocoles de routage basés sur la structure du réseau sont classifiés en trois catégories : protocoles à plat (*Flat based routing*), protocoles hiérarchiques (*Hierarchic based routing/Clustering based routing*) et protocoles basés sur la localisation géographique (*Location based routing*).

### II.2.1.1 Routage à plat

Dans ce type de protocole, tous les nœuds ont le même rôle, homogènes et communiquent entre eux sans aucun autre intermédiaire, et seul la station de base est chargée de la collecte de données issues des différents capteurs. Les topologies plates sont caractérisées par la simplicité des algorithmes utilisés pour le routage des données et la scalabilité du réseau, du fait que les nœuds ont besoin uniquement des informations sur leurs voisins directs afin de participer à la tâche de routage.

### II.2.1.2 Routage hiérarchique

Dans cette architecture le réseau est partitionné en sous ensembles appelés clusters, afin de faciliter la gestion du réseau ainsi que le routage de données. Dans ce type de protocole, chaque cluster est constitué d'un nœud simple et d'un nœud leader (appelé aussi cluster-head). Seul le cluster-head communique avec les autres capteurs ou avec la station de base. Tous les capteurs d'un cluster envoient les données au cluster-head. Ensuite ce dernier s'en charge de les acheminer vers d'autre cluster-head ou bien vers la station de base.

Le choix du cluster-head se fait soit à tour de rôle, soit selon le nombre de voisins en considérant comme cluster-head le nœud ayant le maximum de voisins, soit selon le niveau d'énergie du nœud. Les nœuds à énergie élevée (cluster-head) sont utilisés pour traiter, agréger et acheminer les informations vers la destination ; alors que les membres du cluster se chargent de la tâche de capture. Avec cette méthode on assure une minimisation efficace d'énergie. [1]

### II.2.1.3 Routage basé sur la localisation

Dans ce type de routage, les nœuds sont adressés à l'aide de leurs positions. Les informations sur la position de chaque nœud dans le réseau est nécessaire pour estimer l'énergie consommée lors des différentes émissions. Comme les nœuds capteurs sont déployés dans une région d'une manière aléatoire, les informations de localisation de ces nœuds peuvent être utilisées dans le routage des données d'une manière efficace en termes d'énergie dans le but de maximiser la durée de vie du réseau.

## II.2.2 Classification selon les fonctions du protocole

Les protocoles de routage peuvent être classifiés selon leurs fonctionnalités en quatre catégories : routage basé sur la Qualité de Service (Quality of Service based routing), routage basé sur le flux de données dans le réseau (*Network flow based routing*), routage basé sur des multi-chemins (*Multi-path based routing*), et routage basé sur la négociation (*Negotiation based routing*).

### II.2.2.1 Routage basé sur la Qualité de Service

Dans les protocoles de routage basés sur la QoS, le réseau doit assurer un équilibre entre la consommation d'énergie et la qualité de données. En particulier, le réseau doit satisfaire une certaine métrique de QoS. Ce type de protocole est utilisé dans des applications qu'ont des exigences temps réel (monitoring médical, application militaires, ...etc.).

### II.2.2.2 Routage basé sur le flux de données dans le réseau

Dans ce type de protocole, la phase d'établissement de route est modélisée et résolue comme un problème de demande de flux de données ou le flot représente la route que les paquets empruntent, et la demande représente le taux auquel les paquets sont produits par les différents nœuds.

### II.2.2.3 Routage multi-chemin

Au lieu d'utiliser un seul chemin, les protocoles basés sur ce type de routage maintiennent plusieurs chemins, dans le but d'augmenter les performances du réseau. Lorsque le chemin primaire est défaillant, les données vont être acheminées vers la destination via des chemins alternatifs.



#### II.2.2.4 Routage basé sur la négociation

Ces protocoles utilisent des descriptions de données (métadonnées) afin d'éliminer les transmissions de données redondantes, par une négociation préalable entre la source et la destination. Cette procédure garantit que seules les informations utiles qui seront transmises.

### II.2.3 Classification selon l'établissement de la route

Suivant la manière de la création et la maintenance de routes lors de l'acheminement des données, les protocoles de routage peuvent être classés en trois catégories : Les protocoles proactifs, les protocoles réactifs et les protocoles hybrides.

#### II.2.3.1 Protocoles proactifs

Les protocoles de routage proactifs essaient d'établir les meilleurs chemins existants vers toutes les destinations possibles au niveau de chaque nœud du réseau, tel que chacun d'eux maintient une vue globale de la topologie qui est mis à jour périodiquement à travers l'échange des messages de contrôle. Ceci induit un contrôle excessif surtout dans le cas des réseaux à forte densité. De plus, ils présentent un autre inconvénient dû à la sauvegarde des routes même si elles ne sont pas utilisées.

#### II.2.3.2 Protocoles réactifs

Contrairement au routage proactif, les protocoles de routage réactifs déterminent les routes au moment où une transmission doit être réalisée (à la demande). Lorsque le réseau a besoin d'une route, une procédure de découverte de route est lancée. Une fois la route n'est plus utilisée, elle sera immédiatement détruite ce qui permet une conservation d'énergie. Cependant, le routage à la demande induit une lenteur à cause de la durée nécessaire pour la recherche des chemins, ce qui peut dégrader les performances des applications interactives.

#### II.2.3.3 Protocoles hybrides

Les protocoles hybrides combinent les deux idées des protocoles proactifs et réactifs. Ces protocoles appliquent la politique proactive à l'intérieur de la zone de voisinage ; au-delà de cette dernière, une politique réactive est mise en place pour chercher les routes.

### II.2.4 Classification selon l'initiateur des communications

La communication dans un réseau de capteurs peut être lancée par les nœuds sources ou bien par les nœuds destinataires.

#### II.2.4.1 Communication lancée par la source

Dans les protocoles de communication lancée par la source, les nœuds envoient des données à la destination quand une capture a eu lieu. Ces protocoles sont utilisés dans les applications orientées temps (*time driven*), et orientées événement (*event driven*). Ceci signifie

que les données sont envoyées à certains intervalles ou quand les nœuds capturent certains évènements. [23]

#### II.2.4.2 Communication lancée par la destination

Les protocoles de communication initiée par la destination utilisent le modèle orientées requêtes (*query driven*), dans lesquelles la destination envoie des requêtes décrivant son besoin. A la réception d'une demande, le nœud capteur se charge de répondre par une transmission des données, s'il y a lieu, sinon une diffusion de cette demande aura lieu à tous les nœuds d'une région donnée. [1]

## II.3 Quelques protocoles de routage pour les RCSFs

### II.3.1 Inondation(Flooding)

Le flooding [12, 31] ou inondation en français est une technique classique pour le routage dans les RCSFs. Dans cette approche, chaque nœud recevant un paquet l'envoie à tous ses voisins. Ce processus ne se termine que si le paquet est arrivé à destination ou si le nombre maximum de saut est atteint. Ce protocole n'a nul besoin ni de maintenir une table de routage, ni de découvrir son voisinage et maintenir une topologie bien précise. Par contre, il présente plusieurs inconvénients, entre autres (figure II.2 [31]) :

- *L'implosion* : elle parvient si des messages dupliqués sont envoyés au même nœud.
- *Le recouvrement* : si deux nœuds observant la même région sont sollicités au même instant, leurs nœuds voisins recevront des messages dupliqués.
- *Ignorance de ressource (blindness ressource)* : la technique d'inondations ne tient pas compte des ressources énergétiques disponibles. Un protocole efficace en consommation d'énergie, doit prendre en compte la quantité d'énergie à leur disposition à chaque instant.

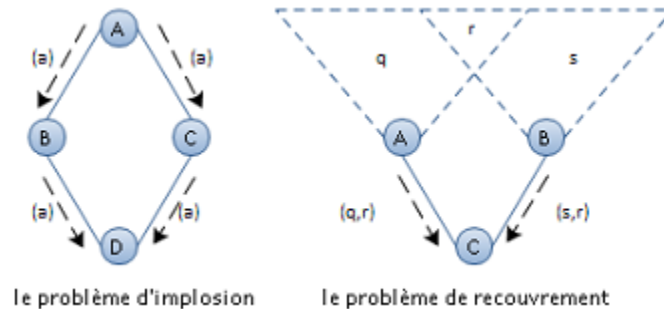


FIGURE II.2 – Problèmes de routage par inondation.

### **II.3.2 Gossiping**

Gossiping [12, 31] est une version modifiée du Flooding, dans laquelle un nœud ne diffuse pas les messages reçus à tous ses voisins, mais il les transmet à un seul, sélectionné aléatoirement. Cette technique évite le problème d'implosion en ayant une seule copie de message au niveau de chaque nœud, mais elle génère un retard dans la propagation des messages dans tout le réseau.

### **II.3.3 Diffusion dirigée (Direct Diffusion)**

Le protocole de diffusion dirigée proposé dans [5] est un protocole important dans le routage data-centric (routage à plat) dans les RCSFs. L'idée de base vise à répandre des données aux nœuds capteurs en utilisant un schéma de nommage pour les données.

La diffusion dirigée suggère l'utilisation de paires attribut-valeur pour les données et les requêtes des capteurs. Le nœud puits (SB) diffuse son intérêt, qui présente une description des tâches affectées à tous les nœuds capteurs du réseau. L'intérêt est basé également sur la présentation attribut-valeur pour désigner par exemple le type de données requises et le débit de transmission vers la SB (nombre d'événements ou de paquets de données envoyé par unité de temps). Durant le processus de dissémination d'intérêt, des gradients sont établis, afin de déterminer les routes qui peuvent être employées ultérieurement dans la collecte des événements capturés. L'une de ces routes est par la suite renforcée pour être utilisée plusieurs fois. Les messages d'intérêt reçus par les nœuds sont enregistrés dans un cache et utilisés par la suite pour comparer les données reçues (qui se propagent) avec les valeurs d'intérêts.

Une entrée d'intérêt dans le cache est composée de plusieurs champs de gradient. Un gradient est un lien réponse avec un voisin dont l'intérêt a été suscité et qui est caractérisé par le débit, la durée et la date d'expiration de données.

L'utilisation des intérêts et les gradients permettent la création des routes entre les destinations et les sources. Plusieurs routes peuvent être établies de telle sorte que l'une d'elle est choisie par renforcement. La destination renvoie le message d'intérêt initial à travers la route choisie. Un intervalle plus petit renforce donc le nœud source sur ce chemin pour envoyer des données plus fréquemment (figure II.3 [6]).

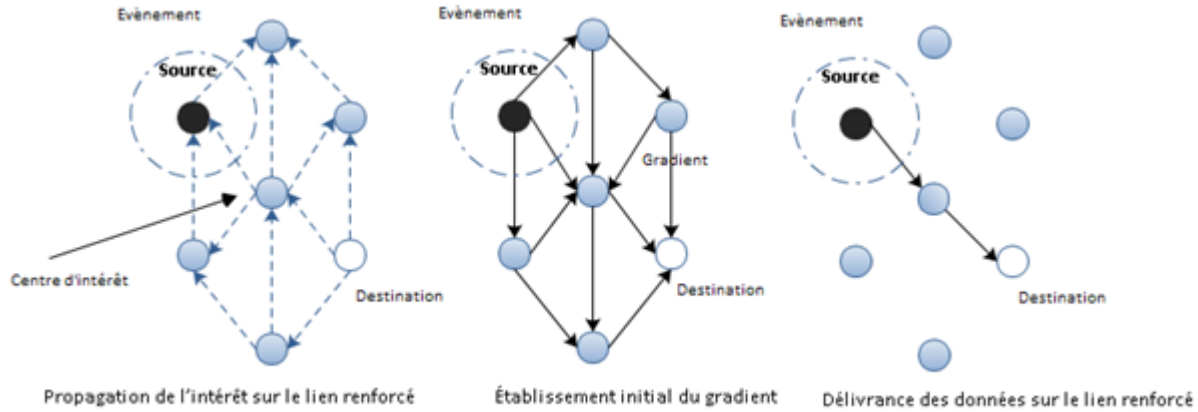


FIGURE II.3 – Les phases du routage de la Diffusion Dirigée

### II.3.4 SPIN : *Sensor Protocols for Information via Negotiation*

Heinzelman et al [13] ont proposé une famille de protocoles appelée SPIN (*Sensor Protocols for Information via Negotiation*), reposant sur un modèle de négociation. Il présente de nombreuses similitudes par rapport à la diffusion dirigée. Il est efficace dans la réduction des données redondantes et économise l'énergie, par rapport aux méthodes de diffusion traditionnelles telles que l'inondation [11], pour pallier aux problèmes de ces méthodes, SPIN adopte deux principes :

- *La négociation* : SPIN précède l'émission d'une donnée par sa description, en utilisant la notion de métadonnées. Le récepteur aura le choix par la suite d'accepter la donnée ou non.
- *L'adaptation aux ressources* : d'une manière continue, les nœuds contrôlent leur niveau d'énergie. Le protocole SPIN accommode son exécution suivant l'énergie restante du capteur, et modifie en conséquence le comportement du nœud.

SPIN utilise particulièrement trois types de paquets à savoir ADV (*ADVertissing*), REQ (*REQuête*) et DATA (*Information*). Les communications dans SPIN se font en trois étapes :

- Lorsqu'un nœud veut émettre une donnée, il émet d'abord un message ADV contenant une description de la donnée en question.
- Un nœud recevant un message ADV, consulte sa base d'intérêt. S'il est intéressé par cette information, il émet un message REQ vers son voisin.
- En recevant un message REQ, l'émetteur transmet à l'intéressé la donnée sous forme d'un message DATA.

Lorsqu'un nœud s'aperçoit que son énergie est descendue sous un certain seuil, il change son mode de fonctionnement, et ne répond à aucun message ADV. La figure suivante illustre ces trois étapes :

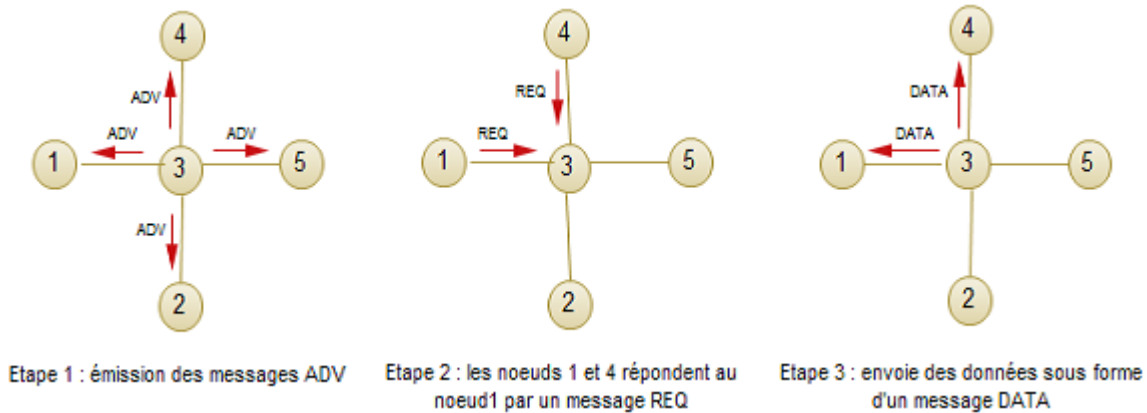


FIGURE II.4 – Un exemple d'exécution du protocole SPIN.

### II.3.5 Routage par rumeur

Le routage par rumeur [7] est principalement destiné pour des applications où le routage géographique n'est pas faisable.

Il part du principe que certains nœuds connaissent des événements que d'autres veulent connaître : quelqu'un demande une info à quelqu'un d'autre mais sans savoir qui ni où il est. Mais faut-il diffuser les événements ou les demandes ? S'il y a beaucoup d'événements, en les diffusant on risque de diffuser des infos qui n'intéressent personne. Mais s'il y a beaucoup de demandes il vaut mieux diffuser les événements que les demandes. La solution retenue : le nœud origine d'un événement l'envoie à un des ses voisins au hasard tel que chaque nœud qui les reçoit fait de même (figure II.5). Chaque nœud qui reçoit l'événement connaît le nœud d'origine de cet événement et le chemin qu'il a parcouru et en garde trace. On diffuse de la même façon les demandes. Quand une demande arrive sur un nœud on regarde si l'on a, sur ce nœud, une trace de l'événement qu'elle demande. Si c'est le cas on sait d'où vient l'événement et nous avons le chemin pour atteindre son nœud d'origine.

Les auteurs ont utilisé un procédé probabiliste, reposant sur le fait suivant : des simulations basées sur la méthode de Monte-Carlo <sup>1</sup> ont montré que la probabilité que deux lignes se croisent au sein d'une région rectangulaire est 0.69. Par conséquent, si nous considérons le puits et la source comme deux points, et en établissant un nombre réduit de mi-chemins depuis la source et le puits, nous aurons une forte chance que deux mi-chemins se joignent, créant ainsi un chemin complet entre la source et la destination, tout en évitant l'inondation [28]. La création de ces mi-chemins se base sur la notion d'agent. Un agent est un paquet

<sup>1</sup>La méthode de Monte Carlo peut être définie comme toute technique numérique de résolution de problèmes au moyen d'un modèle stochastique dans lequel nous utilisons des nombres aléatoires [27].

avec une grande portée qui traverse le réseau de nœud en nœud afin d'établir des tables de relais. Il existe deux types d'agents : d'évènement et de requête (figure II.5 [7]).

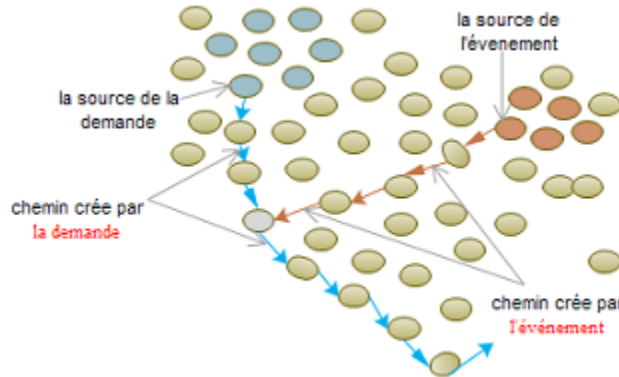


FIGURE II.5 – Un exemple de routage par rumeur.

### II.3.6 GEAR : *Geographic and Energy-Aware Routing*

Le protocole de routage GEAR (*Geographic and Energy Aware Routing*) [30] a été suggéré par Y. Yu et al. Il consiste à utiliser l'information géographique lors de la diffusion des requêtes aux régions cibles car les requêtes contiennent souvent des données géographiques. L'idée est de restreindre le nombre de données dans la diffusion dirigée en prenant en considération uniquement une certaine région, plutôt que d'envoyer les données à l'ensemble du réseau.

GEAR se comporte comme un protocole de routage sur demande. La découverte de la route est lancée seulement quand un nœud veut transmettre des données, aussi aucune mise à jour périodique ou le maintien des tables de routage n'est effectué. Chaque nœud connaît sa propre localisation, son niveau d'énergie, la localisation de chacun de ses voisins et leurs niveaux d'énergie par l'échange des messages d'information (échange des messages Hello). Un chemin existe entre deux nœuds si chacun d'eux a un niveau d'énergie suffisante pour échanger les paquets entre eux, dans le cas contraire c'est un trou.

Avec le protocole GEAR, chaque nœud maintient le coût pour atteindre la destination en passant par ses voisins. Ce coût est divisé en deux parties : un coût estimé et un coût d'apprentissage. Le coût estimé est une combinaison de l'énergie résiduelle et de la distance jusqu'à la destination. Le coût d'apprentissage est un raffinement du coût estimé qu'un nœud dépense pour le routage autour des trous dans le réseau.

### II.3.7 LEACH : *Low Energy Adaptive Clustering Hierarchy*

*W.Heinzelman* et al proposent dans [26] le protocole LEACH comme un nouveau protocole de routage efficace en consommation d'énergie pour les réseaux de capteurs. L'idée proposée par LEACH a été une inspiration pour de nombreux protocoles de routage

hiérarchique, bien que certains protocoles aient été développés de manière indépendante [29].

LEACH est un protocole basé sur les clusters, dans le quel les cluster-head élus collectent les données à partir de tous les nœuds capteurs appartenant à leur cluster, agrègent les données rassemblées par des procédures de fusion, et transmettent ces données directement à la station de base. Les cluster-head élus demeurent ainsi pour une période de temps appelée « round ». Au début de chaque round, chaque nœud détermine la possibilité d'être un cluster-head pendant la période en cour, s'il décide de l'être, il annonce sa décision à tous ses nœuds voisins. Les autres nœuds qui décident de ne pas être un cluster-head, se joignent à l'un des clusters-head élus après avoir reçu leurs décisions, la sélection du cluster-head adéquat se base sur plusieurs paramètres prédéterminés tel que le rapport signal/bruit (SNR).

Dans le protocole LEACH, tous les nœuds sont considéré statiques, homogènes, ils captent leur environnement d'une manière continue, ont un taux de données fixe à transmettre, de plus, les canaux de communication radios sont supposés symétriques.

Les caractéristiques principales de l'algorithme LEACH se résument en : la coordination locale pour l'établissement des clusters et leurs fonctionnement, la rotation aléatoire des clusters-head, et la fusion locale des données pour réduire le taux de communications avec la station de base.

### II.3.8 PEGASIS : *Power-Efficient Gathering in Sensor Information System*

PEGASIS [24] est une version améliorée de LEACH. L'objectif principal de PEGASIS est de former une chaîne entre les nœuds de sorte que chaque nœud communique uniquement avec ses voisins les plus proches. Les données collectées sont transmises d'un nœud à un autre qui les agrègent jusqu'à ce qu'elles arrivent à un nœud particulier (leader) qui les transmet à la station de base. Les nœuds qui transmettent les données à la station de base, sont choisis tour à tour selon une politique round-robin dans le but de réduire l'énergie moyenne dépensée par un nœud durant une période (round). Dans ce protocole, au lieu de former des clusters, PEGASIS procure à un seul nœud dans la chaîne l'envoi de données à la station de base.

Les résultats de simulation ont montré que PEGASIS est plus préformant en termes de réduction de la consommation d'énergie, ce qui lui permet de prolonger de deux à trois fois la durée de vie d'un réseau de capteurs par rapport au protocole LEACH. Ces performances sont obtenues grâce à l'élimination du surcoût causé par le processus de formation de clusters dans LEACH, et par la réduction du nombre de transmissions et de réceptions en employant l'agrégation de données. PEGASIS exige toujours un ajustement dynamique de la topologie puisqu'un nœud devrait connaître le niveau d'énergie de ses voisins avant de relayer ses données. Cependant, un tel ajustement de la topologie pourrait causer un surcoût important. En outre, PEGASIS suppose que chaque nœud à la possibilité de transmettre les données capturées à la station de base directement ou à travers un nœud intermédiaire.

D'autre part, PEGASIS suppose que tous les nœuds maintiennent une table contenant

les localisations de tous les autres nœuds dans le réseau. En résumé, PEGASIS est adapté seulement aux capteurs sans fil dont les nœuds sont immobiles. Son évaluation dans des environnements mobiles pourrait dégrader considérablement ses performances.

### II.3.9 TEEN : *Threshold sensitive Energy Efficient sensor Network protocol* et APTEEN : *Adaptive Periodic Threshold-sensitive Energy Efficient Sensor Network Protocol*

TEEN est un protocole de routage basé sur les clusters proposé par *Manjeshwar et al.* [3]. Son comportement est très semblable au protocole LEACH. Ce dernier est destiné aux applications time-driven (*orienté temps*). Dans ce type d'application, la donnée est propagée d'une manière périodique. Cependant, ce genre de protocole est inadapté pour les applications event-driven (*orienté événements*), où un comportement réactif est nécessaire pour le bon fonctionnement du système. TEEN a été développé pour améliorer LEACH afin de répondre aux exigences des applications *event driven*.

TEEN est conçu pour être sensible à des changements soudains des attributs tels que la température. Il utilise la même stratégie que LEACH pour l'étape de formation des clusters, mais adopte une approche différente pour la phase de transmission des données. Durant cette étape, il utilise deux paramètres définis par l'utilisateur appelé *hard threshold (ht)* et *soft threshold (st)* et ceci, pour pouvoir déterminer le besoin de transmission de la donnée captée vers la station de base. Si la valeur captée excède *ht* pour la première fois, elle est stockée dans une variable et transmise durant le temps alloué par le chef du cluster au nœud concerné. Si la valeur captée dépasse, par la suite, la valeur stockée par une magnitude *st*, le nœud décide de la transmettre et stocke cette nouvelle valeur dans son cache pour les comparaisons ultérieures.

Pour remédier aux limitations du protocole TEEN, les auteurs ont proposé une extension de TEEN appelée APTEEN [4]. Ce dernier est une extension de TEEN qui fait à la fois la collection des captures périodique de données et qui réagit aux événements critiques. Quand la station de base forme des clusters, les clusters head diffusent les attributs, les valeurs des seuils, ainsi que le calendrier de transmission à tous les nœuds. Le cluster-head effectue également l'agrégation de données afin d'économiser l'énergie.

Les protocoles TEEN et APTEEN conviennent pour les applications critiques. Dans les deux protocoles, le facteur clé est la valeur de l'attribut mesuré. La caractéristique supplémentaire d'APTEEN est la capacité de changer la périodicité et les paramètres de TEEN en fonction des besoins des utilisateurs et des applications.

### II.3.10 SAR : *Sequential Assignment Routing*

Le routage SAR proposé dans [8] est l'un des premiers protocoles dans les RCSFs qui a introduit la notion de QoS dans les décisions de routage. La sélection du chemin dans le protocole SAR dépend de trois facteurs : les ressources énergétiques, les métriques de qualité



de service (QoS) de chaque route candidate, ainsi que le niveau de priorité des paquets. Le protocole utilise une approche multi-chemins pour éviter les messages de contrôle qui peuvent résulter de la réestimation des chemins quand celles employées est défaillantes, dès lors, un schéma de restauration de chemins est utilisé.

Pour créer des chemins multiples à partir de chaque nœud jusqu'au nœud puits, plusieurs arbres d'une longueur d'un saut de ce nœud sont initialement construits, chaque arbre est par la suite développé en allant successivement du nœud puits vers les autres nœuds, tout en évitant ceux qui ont les valeurs faible de QoS ( faible débit, grand délai de transmission) et d'énergie disponibles. A la fin du processus, chaque nœud fera partie des chemins multiples, et le capteur pourra savoir le nœud voisin qui peut être utilisé pour transmettre le message. Dans l'algorithme SAR, deux paramètres sont associés à chaque chemin pour chaque nœud, une métrique QoS additive, en plus d'une valeur qui mesure la ressource énergétique disponible dans chaque nœud, cette dernière est calculée via l'estimation du nombre de paquets pouvant être routés suivant un chemin donnée, sans épuisement de l'énergie disponible chez le nœud. SAR calcule par la suite une métrique pondérée de QoS égale au produit de la métrique QoS additive par un coefficient lié au niveau de priorité du paquet transmit.

### II.3.11 SPEED : *A stateless protocol for real-time communication in sensor networks*

SPEED est un protocole proposé dans [25], afin d'assurer certaines métriques de qualité de service lors du routage de données dans les RCSFs. Ils ont ajouté une métrique supplémentaire par rapport à GEAR : le délai. Ce protocole exige que chaque nœud maintienne des informations sur ses voisins. Il procède par un routage géographique pour sélectionner les nœuds de prochains sauts jusqu'à l'arrivée à la destination finale(le nœud puit). En outre, SPEED assure une vitesse de livraison des paquets constante, que nous notons *Setspeed*. Ceci permet de garantir des délais de livraison de bout en bout acceptables. Ces délais peuvent être estimés en divisant la distance séparant le nœud source du puit par la vitesse *setspeed*. Ainsi, chaque application peut estimer, avant l'utilisation du protocole, les délais de bout-en-bout offerts, en divisant la distance qui sépare les capteurs sources du capteur puits par la vitesse des paquets. SPEED englobe également des techniques qui permettent d'éviter la congestion dans le RCSF.

## II.4 Synthèse des protocoles de routage dans les RCSFs

### II.4.1 Classification

Après avoir cité quelques protocoles de routage utilisés dans les réseaux de capteurs sans fil, la table II.1 récapitule les différentes classes associées à chaque protocole ; il indique ces derniers selon leur appartenance à une ou plusieurs classes.

Lors de la classification de ces protocoles selon les approches de routage étudiées, nous remarquons que ces protocoles peuvent appartenir au plus d'une catégorie. A titre d'exemple,

Classification Protocoles	Structure du réseau	Fonction du protocole	Etablissement de la route	Initiateur de la communication
Flooding	A plat	Non	Réactif	Source
Gossiping	A plat	Non	Réactif	Source
SPIN	A plat	Multi-chemin, Négociation	Proactif	Source
Direct Diffusion	A plat	Multi-chemin, Négociation	Réactif	Destination
Rumor Routing	A plat	Non	Réactif	Source
GEAR	Localisation	Non	Réactif	Destination
SPEED	Localisation/A plat	QdS	Hybride	Source
SAR	A plat	QdS, multi-chemin et négociation	Hybride	Source
LEACH	Hiérarchique	Non	Proactif	Source
TEEN	Hiérarchique	Non	Réactif	Destination
APTEEN	Hiérarchique	Non	Hybride	Source
PEGASIS	Hiérarchique	Non	Hybride	Source

TABLE II.1 – Classification des différents protocoles de routage dans les RCSFs

Direct diffusion qui est un protocole à plat, Multi-chemin, basé sur la négociation, réactif et avec communication lancée par la destination.

## II.4.2 Comparaison

La table II.2 inspiré des travaux [31, 22] illustre la comparaison entre les protocoles étudiés précédemment suivant les critères cités ci-dessous :

- **L'agrégation des données** : c'est un nouveau paradigme pour le routage dans les réseaux de capteurs. L'idée de base consiste à combiner les données provenant de différents nœuds sources en éliminant les redondances existantes et en minimisant aussi le nombre de transmissions possibles pour économiser la quantité d'énergie consommée.
- **La mobilité** : un autre paramètre qui semble intéressant à prendre en considération, lors de comparaison entre ces protocoles de routage, est la mobilité des nœuds. Cependant, il existe des situations ou des environnements où les nœuds capteurs doivent être mobiles. Dans ce cas, la mise à jour fréquente de la position du nœud et la propagation de cette information dans le réseau peut considérablement diminuer l'énergie des nœuds.
- **La scalabilité** : la scalabilité d'un protocole de routage est son aptitude à conserver ses propriétés fonctionnelles malgré le changement de la taille du réseau.
- **Efficacité énergétique** : c'est une mesure du rapport des paquets totale délivrée à la destination et l'énergie totale consommée par les nœuds du réseau.

$$\frac{\text{Taux de succès} * \text{Nombre total de paquets envoyés à la station de base}}{\text{l'énergie totale consommée}} \quad (kbit/s)$$

<b>Protocoles</b> \ <b>Critères</b>	<b>Agrégation de données</b>	<b>Mobilité</b>	<b>scalabilité</b>	<b>Efficacité énergétique</b>
Flooding	Non	Non	Bonne	moyenne
Gossiping	Non	Non	Bonne	moyenne
SPIN	Oui	possible	limitée	faible
Direct Diffusion	Oui	limitée	limitée	faible
Rumor Routing	Oui	Très limitée	Bonne	faible
GEAR	Non	Limitée	limitée	moyenne
SPEED	Non	Non	Limitée	faible
SAR	Oui	Non	limitée	faible
LEACH	Oui	SB fixe	Bonne	élevé
TEEN	Oui	SB fixe	Bonne	élevé
APTEEN	Oui	SB fixe	Bonne	très élevé
PEGASIS	Non	SB fixe	Bonne	élevé

TABLE II.2 – Comparaison entre les différents protocoles de routage dans les RCSFs

D'après la table ci-dessus (table II.1), nous remarquons que chaque protocole est bon par rapport à certains critères de comparaison et mauvais par rapport à d'autres. Ce qui nous mène à dire qu'il est difficile de concevoir un protocole qui supporte tous ces critères.

## II.5 Conclusion

Ce chapitre nous a permis de voir les différentes familles de protocoles de routage dans les réseaux de capteurs sans fil, et nous avons donné le principe de base des protocoles les plus répandus. Dans ces derniers la priorité est donnée à la consommation d'énergie.

L'étude de ces protocoles nous a permis de les classifier et de les comparer. Bien que plusieurs stratégies adoptées par ces protocoles de routage paraissent prometteuses, mais il existe toujours certains défis qui persistent et nécessitent leur prise en considération par les protocoles de routage dans les réseaux de capteurs.

Nous nous intéressons par la suite à la conception d'un nouveau protocole de routage économe en consommation d'énergie pour les RCSFs et qui a pour but d'augmenter la durée de vie du réseau.

# Chapitre III

## Protocole de routage pour l'économie d'énergie dans les RCSFs

### III.1 Introduction

La consommation d'énergie est une contrainte importante dans les réseaux de capteurs sans fils. De ce fait, L'épuisement énergétique d'un certain nombre de nœud entraîne un changement significatif sur la topologie globale du réseau, et peut nécessiter un routage de paquets différent et une réorganisation totale du réseau. L'efficacité en consommation d'énergie représente une métrique de performance significative, qui influence directement sur la durée de vie du réseau tout en entier.

Dans ce chapitre nous proposons un nouveau protocole de routage pour les réseaux de capteurs sans fils. Son objectif est d'augmenter la durée de vie du réseau en équilibrant l'énergie de la manière la plus équitable possible. Ce protocole est de type de routage réactif, destiné aux applications adaptées à l'utilisateur suivant le mode de fonctionnement orienté requête avec une structure plate du réseau tel que les capteurs sont équivalents et participent tous au routage de données.

### III.2 Problématique

La gestion d'énergie dans les réseaux de capteurs sans fil est cruciale puisque les nœuds capteurs ont de sévère contraintes d'énergie. Ces nœuds fonctionnent grâce à l'énergie fournie par leur batterie. C'est pourquoi, un des problèmes parmi les plus importants dans ces réseaux consiste à réduire la consommation d'énergie, de sorte à maximiser la durée de vie du réseau. Par conséquent, concevoir des protocoles prenant en considération la consommation d'énergie devient donc un défi majeur pour la mise en place d'un réseau fiable.

Ainsi, l'un des problèmes sur ce type de réseaux est celui de routage qui doit éviter une consommation excessive d'énergie lors du processus d'acheminement des données, car les nœuds sont autonomes et ont chacun un potentiel énergétique limité. A cet égard, utiliser des techniques efficaces optimisant la consommation d'énergie dans le processus de routage

est indispensable pour réaliser des économies d'énergie significatives et prolonger la durée de vie des nœuds et par conséquent, celle du réseau. Ces techniques doivent assurer qu'aucune donnée redondante ne devrait être transférée dans une communication. En outre, le trafic devrait être distribué équitablement parmi les nœuds pour éviter la mort prématurée du réseau.

### III.3 Motivations

Les progrès de ces dernières années en microélectronique ont permis de fabriquer des capteurs de plus en plus petits, de plus en plus performants et avec des autonomies énergétiques grandissantes. L'efficacité dans l'utilisation de ces dispositifs est d'une importance vitale étant donné que ces capteurs fonctionnent dans des régions hostiles et leurs batteries ne peuvent pas être remplacées ou rechargées.

La motivation principale de notre travail réside dans le fait que la station de base possède une grande capacité de stockage et de calcul, d'où l'idée d'exploiter cette propriété pour la conception de notre protocole. La SB est sensé trouver les chemins optimaux vers les capteurs possédant l'information requise afin d'éviter le calcul au niveau des capteurs. Par conséquent, une maintenance de la topologie et une mise à jour des informations relatives à l'énergie résiduelle des capteurs sont nécessaires. En plus, les techniques conçues pour les réseaux ad hoc traditionnels ne sont pas bien adaptées aux réseaux de capteurs, d'où de nombreuses contraintes doivent être résolues pour les réseaux de capteurs sans fil.

### III.4 Estimation de distances avec RSSI

Dans les réseaux de capteurs sans fil, chaque capteur est équipé d'une radio. De ce fait, un nœud à l'écoute d'une transmission radio doit être en mesure d'utiliser la force du signal reçu pour calculer sa distance de l'émetteur. RSSI (*Received Signal Strength Indication*) [14] propose une solution élégante pour l'estimation des distances dans les réseaux de capteurs puisque tous les nœuds possèdent leurs propres dispositifs radio.

Dans notre protocole nous nous intéressons particulièrement à cette méthode afin de mesurer la distance qui sépare deux nœuds capteurs. Elle mesure la puissance du signal au niveau du récepteur et elle est fondée sur l'estimation de la puissance de transmission (communication). L'indicateur de puissance du signal reçu (ou RSSI) est une solution relativement peu coûteuse sans aucun périphérique supplémentaire, sachant que tous les nœuds capteurs sont susceptibles d'avoir l'équipement radio.

### III.5 Modèle énergétique

Le niveau d'énergie d'un capteur est calculé en fonction du nombre de transmissions, réceptions au cours d'une période  $t$  et l'énergie consommée par les opérations de traitement de données. Pour l'évaluation de la consommation d'énergie par un capteur pendant une

période  $t$ , nous avons utilisé le modèle énergétique décrit dans [26]. Dans ce modèle (figure III.1), la consommation d'énergie concerne principalement les transmissions et les réceptions.

*Heinzelman et al.* [26] proposent un modèle radio de consommation d'énergie (figure 1). Ainsi, les énergies nécessaires pour émettre  $ETx(s, d)$  et recevoir  $ERx(s)$  des messages sont données par :

Pour émettre un message de  $s$  bits vers un récepteur loin de  $d$  mètres, l'émetteur consomme :

$$ETx(s, d) = (E_{elec} * s) + (E_{amp} * s * d^2)$$

Pour recevoir un message de  $s$  bit, le récepteur consomme :

$$ERx(s) = E_{elec} * s$$

$E_{elec}$  et  $E_{amp}$  représentent respectivement l'énergie de transmission électronique et d'amplification (figure 1).

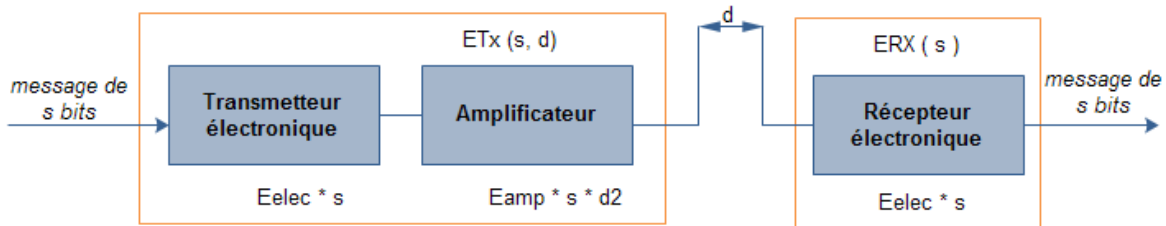


FIGURE III.1 – Modèle de consommation d'énergie.

## III.6 Hypothèses

Avant de présenter notre protocole, nous allons introduire dans cette section quelques hypothèses nécessaires pour le bon fonctionnement de notre protocole.

- Les capteurs sont fixes, leurs positions sont précises et uniques, et la contrainte de mobilité sur les capteurs n'est pas prise en compte.
- Les capteurs sont identiques et à capacités égales (même énergie initiale, même capacité de stockage et de traitement de données) sauf pour le cas de la station de base (énergie illimitée, grande capacité de calcul et de stockage).
- Les liens sont symétriques. Un nœud peut calculer la distance approximative qui le sépare par rapport à un autre nœud, en se basant sur la force du signal reçu.

- Chaque nœud est capable d'estimer son niveau d'énergie à chaque instant.
- La mort de chaque capteur n'est causée que par l'épuisement de son énergie.
- Tous les capteurs collectent les données, et servent une unique station de base.
- Chaque capteur possède un identifiant unique.

## III.7 Description du protocole

Dans cette section, nous présentons les phases principales de notre protocole de routage à basse consommation d'énergie. Notre solution vise à équilibrer les charges et la consommation énergétique des capteurs constituant le réseau. Nous introduisons un nouveau protocole de routage dans lequel la SB a une vue globale sur la topologie du réseau avec les distances cartésiennes existantes entre les différents nœuds. Ce qui lui permet de déterminer le chemin le plus optimal menant à chaque nœud destination et pour le faire nous proposons une méthode qui prend en considération l'équilibrage de la charge du réseau.

Lorsque la SB a déterminé la route optimale menant vers la destination, nous proposons un nouveau mécanisme qui permet de minimiser la taille des paquets en envoyant une seule valeur entière considérée comme le résumé du chemin, au lieu d'attacher tout le chemin dans le paquet à envoyer, chaque nœud déduit le prochain nœud successeur en appliquant une fonction spécifique sur cette valeur. Ce protocole suit le mode de délivrance orienté requête c.-à-d. les capteurs ne réagissent qu'à l'arrivée d'une requête. En effet, vu que la station de base est dotée d'une grande capacité de calcul, de stockage, et d'une énergie illimitée, elle est considérée comme étant l'élément central de ce protocole. Elle est sensée trouver des routes optimales économes en énergie, de sorte que la durée de vie du réseau soit prolongée, et effectuer les traitements et les calculs les plus consommateurs d'énergie afin d'éviter le calcul au niveau des capteurs. Les données capturées sont acheminées par un nœud capteur, depuis son environnement vers la SB en utilisant un chemin déjà choisi par la SB elle-même.

Le principe de ce protocole se résume en trois phases : *la phase d'installation*, *la phase de transmission* et *la phase de maintenance*.

**Phase d'installation** : dans cette phase la SB initie le protocole par la diffusion d'un paquet nommé *INST*, afin de récupérer les informations sur les distances qui séparent chaque nœud de ces voisins, ce qui lui permet d'avoir une vue globale sur la topologie du réseau, et de construire un vecteur d'énergie contenant les niveaux d'énergie résiduelle du chaque capteur. Cette phase est exécutée lors de l'installation du réseau.

**Phase de transmission** : cette phase est exécutée à chaque fois où la SB veut acquérir des informations à partir d'un capteur donné, son objectif est de déterminer les chemins les plus optimaux d'aller et retour vers un nœud destination, et de minimiser au maximum la taille des paquets à envoyer c.-à-d. lorsque la SB veut récupérer des informations à partir d'un capteur précis, elle recense toutes les routes menant à ce capteur, ceci est faisable vu

que la SB connaît la topologie du réseau et l'énergie résiduelle de chaque capteur après l'exécution de la première phase, et elle envoie la requête au capteur en question en lui précisant les routes les plus optimales, en minimisant la taille des paquets à transmettre afin que les capteurs intervenants ne consomment pas trop d'énergie durant le processus de routage.

**Phase de maintenance** : elle s'exécute dans le cas de l'épuisement de l'énergie d'un capteur. Son objectif est de permettre à la SB de connaître la situation du réseau à tout moment en mettant à jour les informations sur la topologie du réseau et le vecteur d'énergie.

### III.7.1 Phase d'installation

Le but visé dans cette phase est la construction d'une matrice qui représente la topologie du réseau et un vecteur d'énergie permettant à la SB d'avoir des informations sur le niveau d'énergie résiduelle de chaque capteur appartenant au réseau.

La matrice de topologie possède une taille  $n \times n$ , tel que  $n$  représente le nombre de nœuds constituant le réseau. La diagonale sera fixée à -1, c'est à dire, un nœud n'est pas un voisin de lui-même. Tel que chaque case de cette matrice correspond à la distance qui sépare un nœud de chacun de ses voisins, si un nœud  $i$  n'est pas voisin d'un nœud  $j$  la case correspondante dans la matrice de topologie aura la valeur *zéro*.

Concernant le vecteur d'énergie, ce dernier possède une taille égale à  $n$ , tel que chaque case de ce vecteur correspond à l'énergie résiduelle de chaque capteur, où l'indice de chaque case représente son identifiant.

La station de base initie le protocole par la diffusion d'un paquet nommé *INST* contenant son identifiant vers tous ses voisins et chaque nœud recevant le paquet *INST* procède comme suit :

- Il estime la distance qu'il le sépare de l'émetteur en se basant sur la puissance du signal reçu en utilisant RSSI (*Received Signal Strength Indication*)[14] vu dans la section III.4.
- Il construit une table de voisinage, telle que chaque entrée dans la table indique l'identifiant de nœud voisin ainsi que la distance qu'il les sépare.
- Il diffuse le paquet *INST* contenant son identifiant à tous ses voisins.

Chaque nœud exécute la procédure de découverte de voisins durant un laps du temps  $t$  qui correspond au temps nécessaire pour avoir une réponse de voisin le plus éloigné. La valeur de  $t$  est calculée à base de la distance maximale que peut atteindre le signal. Cet intervalle du temps permet à chaque nœud de connaître tous ses voisins possibles, et de déterminer l'instant de la transmission de la table de voisinage vers la SB.

A la fin de la diffusion du paquet *INST*, chaque nœud aura une table de voisinage, contenant la liste de ses voisins et les distances qu'il le sépare de chacun d'eux.



Après la construction des tables de voisinage, chaque nœud construit un paquet nommé *repINST* contenant son identifiant et sa table de voisinage puis il le diffuse vers la SB.

Afin de minimiser le nombre de transmissions du paquet *repINST* et d'éviter les boucles, chaque nœud recevant un paquet *repINST* procède comme suit :

- Tous d'abord, il vérifie l'identifiant de nœud source avec la liste des identifiants déjà enregistrés, s'il existe déjà alors il détruit le paquet sinon il enregistre l'identifiant de nœud source du paquet *repINST* ensuite il le transmet à tous ses voisins.
- Ce processus se répète jusqu'à l'arrivée du paquet à la SB.

A la réception de toutes les tables de voisinage, la SB procède à la construction de la matrice de topologie. Sachant que l'énergie initiale des capteurs est connue, et l'énergie consommée lors de la diffusion des paquets *INST* et *repINST* peuvent être calculés vu que la SB connaît la topologie du réseau et la distance qui sépare chaque nœud de chacun de ses voisins, la SB procède au calcul des seuils critiques pour chaque nœud appartenant au réseau et à l'élaboration de vecteur d'énergie en l'initialisant de la manière suivante : nous insérons dans la case correspondante au nœud  $i$  son énergie actuelle (c.-à-d. son énergie après la diffusion des paquets *INST* et *repINST*) qui se calcule en utilisant la formule suivante :

$$EA_i = EI_i - EV_i \dots (1)$$

Telque :

- $EA_i$  : l'énergie actuelle de nœud  $i$ .
- $EI_i$  : l'énergie initiale de nœud  $i$  (avant le déploiement des capteurs, sachant que l'énergie initiale est la même pour tous les capteurs).
- $EV_i$  : l'énergie consommée lors de la diffusion des paquets *INST* et *repINST* par le nœud  $i$ .

A la fin de cette phase, la station de base aura une vue globale de la topologie du réseau ainsi que les distances qui séparent chaque nœud de ses voisins. Elle connaît aussi le niveau d'énergie de tous les nœuds capteurs appartenant au réseau.

Afin de mieux comprendre le fonctionnement de cette phase, nous présentons ci-dessous un exemple illustratif.

Exemple :

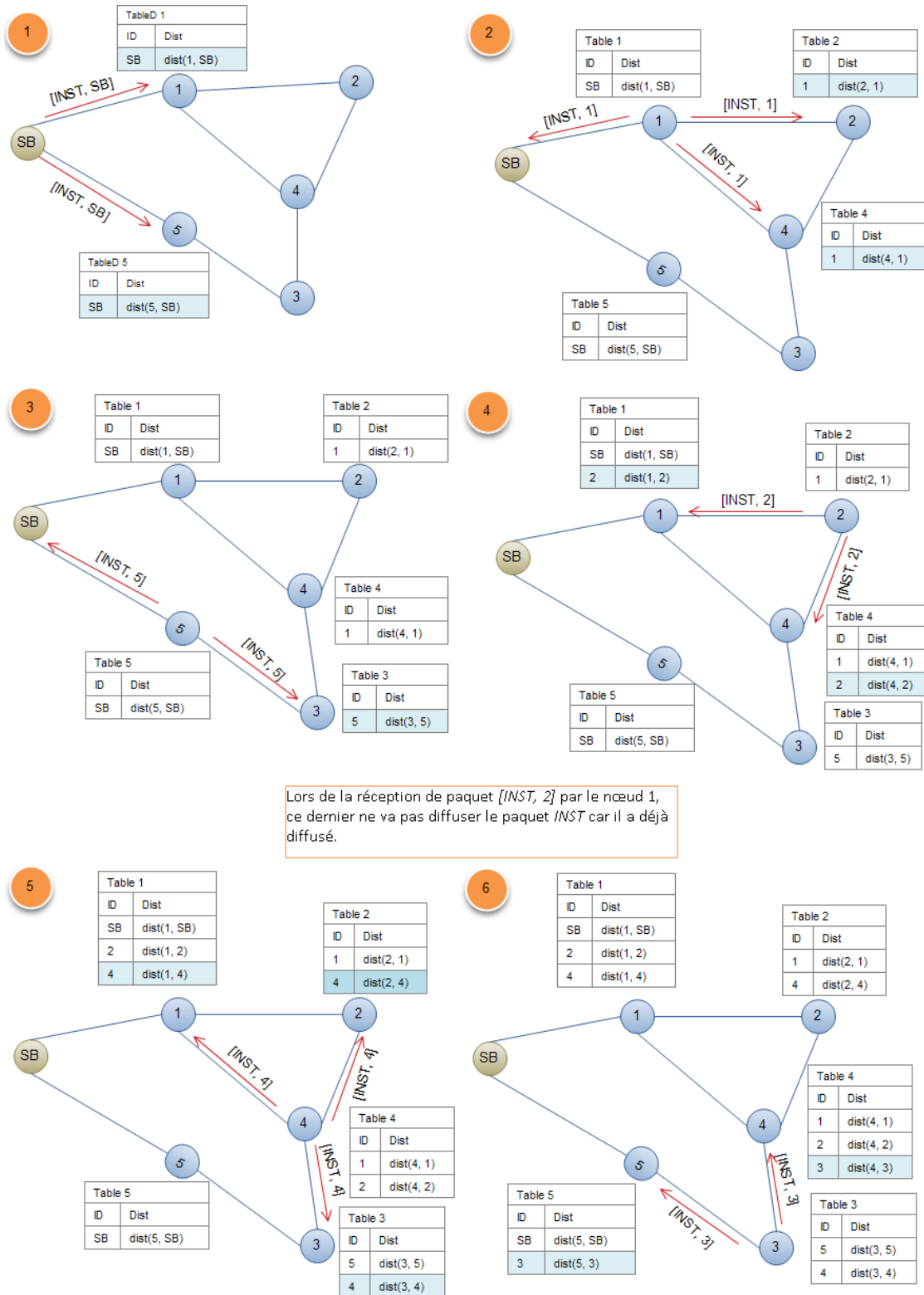
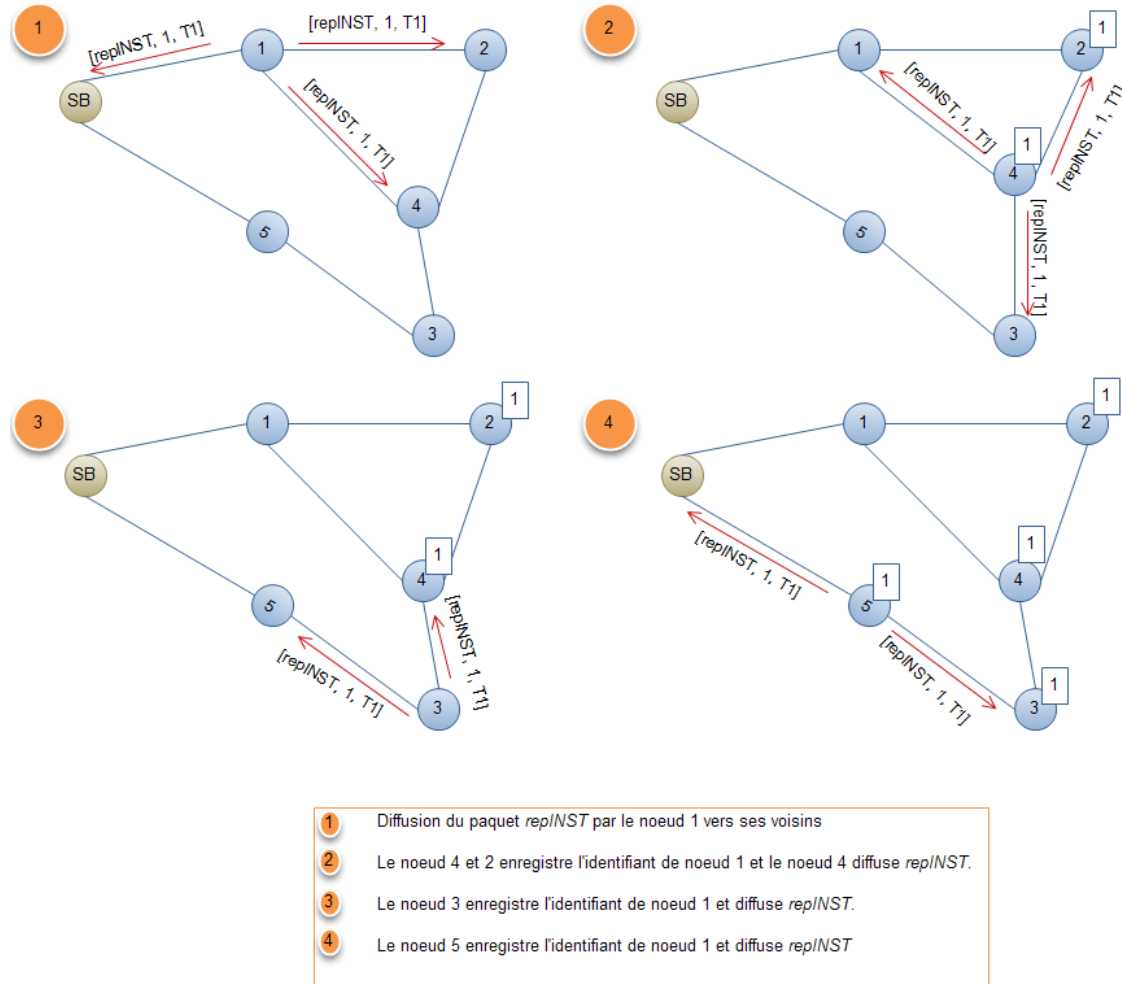


FIGURE III.2 – Un exemple illustratif de la diffusion du paquet  $INST$ .

A la fin de la diffusion de paquet *INST*, tous les nœuds posséderont une table locale indiquant la liste de leurs voisins. La figure suivante décrit le processus de transmission des paquets *repINST* (contenant l'identifiant de l'émetteur et sa table de voisinage  $T_i$ ) vers la SB effectué par le nœud 1.



A la réception de tous les paquets *repINST*, la SB construit la matrice de topologie (fig III.3).

	SB	1	2	3	4	5
SB	-1	dist( SB , 1 )	0	0	0	dist( SB , 5 )
1	dist(1 , SB)	-1	dist(1 , 2)	0	dist(1 , 4)	0
2	0	dist( 2 , 1)	-1	0	dist(2 , 4)	0
3	0	0	0	-1	dist(3 , 4)	dist(3 , 5)
4	0	dist(4 , 1)	dist(4 , 2)		-1	0
5	dist(5 , SB)	0	0	dist(5 , 3)	0	-1




FIGURE III.3 – La matrice de topologie

En effet, cette solution de base n'est pas optimale, à cause de nombre important de transmissions inutiles, de plus la SB reçoit plusieurs paquets *respINST* venant d'un même nœud (redondance). Ce qui engendre une consommation excessive d'énergie surtout dans le cas où la taille des tables de voisinage est importante.

Pour remédier à ces inconvénients, nous proposons une amélioration qui consiste à envoyer le paquet *repINST* sur le chemin optimal, en termes de distance, menant à la SB, de sorte que chaque nœud appartenant à ce chemin le transmet au voisin le plus proche de la SB.

Pour adapter cette solution à notre protocole, nous préservons l'idée de la diffusion du paquet *INST* et la façon de construire les tables de voisinage, sauf que nous envisageons la solution suivante pour remédier aux inconvénients cités ci-dessus.

Nous avons ajouté un champ  $d$  (la distance qui sépare un nœud de la SB) au paquet *INST*. Tel que la structure de ce dernier a les champs suivants :

- *INST* : lancer la procédure de découverte de voisins.
- *ID* : identifiant du nœud émetteur.
- *D* : la distance qui sépare le nœud dont l'identifiant est *ID* de la SB.

En appliquant cette solution, la phase d'installation est décrite comme suit : la SB initie la phase d'installation par l'envoi du paquet *INST* (*INST*, 0, SB). Chaque nœud recevant le paquet d'installation :

- Estime la distance qui le sépare de l'émetteur en utilisant l'indicateur de puissance du signal reçu (ou RSSI).
- Met à jour sa table de voisinage.
- Enregistre la distance minimale *DistMin* qui le sépare de la SB, qui se calcule de la manière suivante :

$DistMin = \min (DistMin \text{ enregistrée précédemment, la nouvelle distance } d \text{ calculée}).$   
*Tel que :  $d = \text{la distance estimée en utilisant RSSI} + \text{la distance reçue}.$*

- Lance la procédure de découverte de voisins en diffusant le paquet *INST* contenant la distance minimale enregistrée (voir Algorithme 1).

---

**Algorithm 1** Diffusion du paquet *INST*

---

```

1:  $DistMin = \infty$ ;
2:  $j \leftarrow 0$ ;
3: pour (chaque nœud intermédiaire  $N_i$  recevant le paquet INST) faire
4:    $DistEmetteur \leftarrow$  la distance qui sépare  $N_i$  de l'émetteur;
5:    $tableVoisinage[j][0] = IDN_i$ ; //  $IDN_i$  : identifiant de nœud  $N_i$ 
6:    $tableVoisinage[j][1] = DistEmetteur$ ;
7:    $d = DistEmetteur + DistReçu$ ;
8:    $DistMin = \min (DistMin, d)$ ;
9:   Diffuser le paquet INST en insérant dans le champ D la valeur  $DistMin$ ;
10:   $j \leftarrow j + 1$ ;
11: fin pour

```

---

A la fin de la diffusion du paquet *INST*, chaque nœud aura à son niveau une table de voisinage, ainsi que le voisin le plus proche de la SB.

Pour acheminer les paquets *repINST* vers la SB, chaque nœud recevant ce dernier le route vers le voisin dont la distance enregistrée est minimale.

A fin de mieux comprendre la solution un exemple illustratif est présenté ci-dessous :

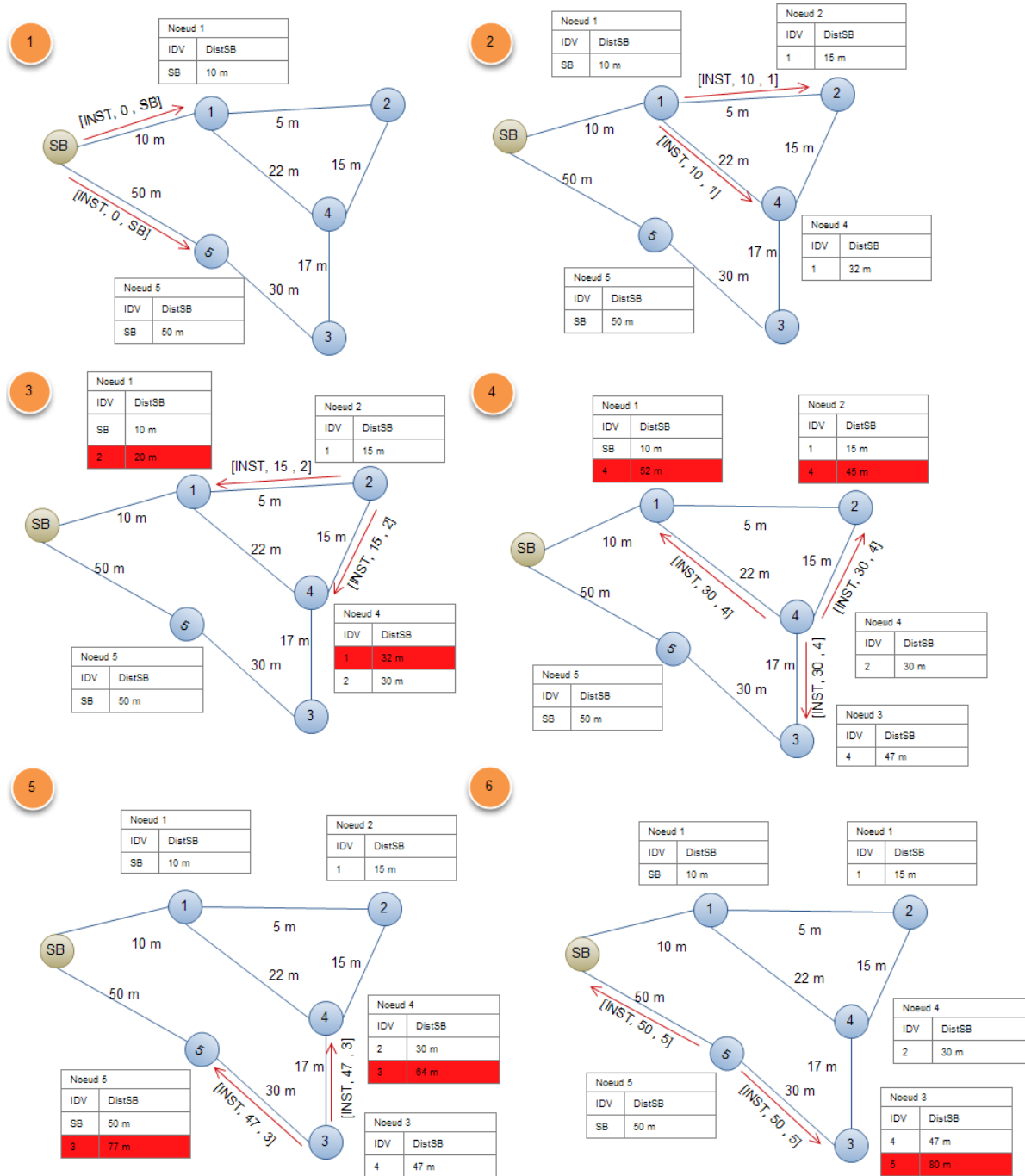


FIGURE III.4 – Un exemple illustratif de la diffusion du paquet *INST*

La figure ci-dessous représente l'envoi du paquet *repINST* vers la SB :

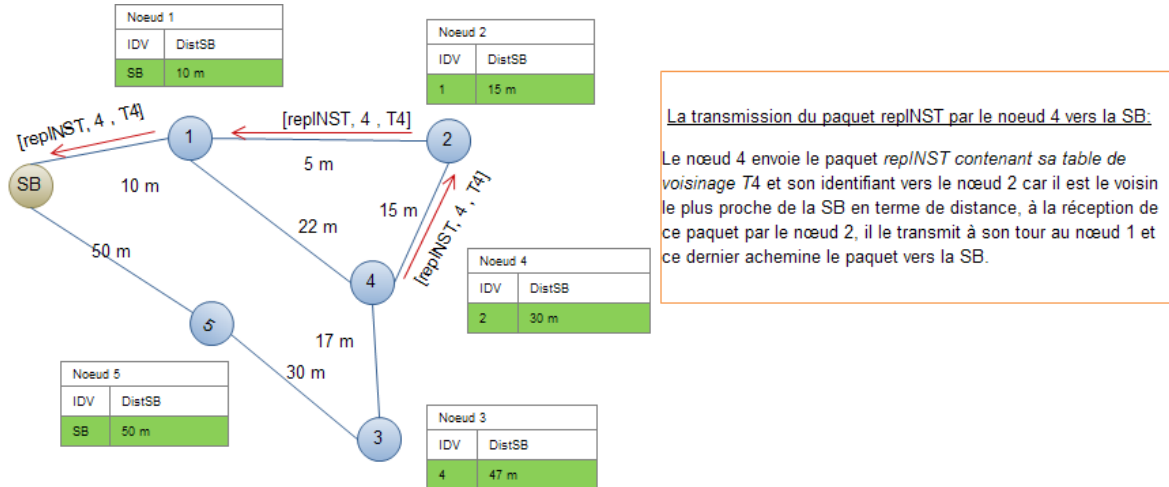


FIGURE III.5 – Exemple illustratif de l'envoi du paquet *repINST*

A la fin de la construction des tables de voisinages chaque nœud doit acheminer sa table vers la SB. La table de voisinage d'un nœud représente les distances qui séparent ce dernier de tous ses voisins, dans le cas où un nœud a plusieurs voisins la taille de sa table devient importante, cela provoque une consommation élevée d'énergie lors de sa transmission vers la SB. Pour pallier à ce problème, nous proposons de minimiser la taille des tables de voisinage de la manière suivante :

Puisque les liens radio sont symétriques, avoir l'information que le nœud  $i$  est un voisin de nœud  $j$  est suffisant pour dire que le nœud  $j$  est aussi un voisin du nœud  $i$ . Par conséquent, chaque nœud recevant le paquet *repINST* consulte la table de voisinage contenant dans ce paquet, et il supprime la ligne qui indique qu'il est voisin de ce nœud, puis il le transmet vers le voisin le plus proche de la SB (Voir Algorithme 2).

---

**Algorithm 2** Minimisation de la taille des tables de voisinages envoyées

---

- 1: **pour** (chaque nœud  $N_i$  recevant le paquet *repINST*) **faire**
  - 2:     *consulter la table de voisinage contenant dans le paquet *repINST* ;*
  - 3:      $n \leftarrow$  *taille de la table de voisinage ;*
  - 4:     **pour** ( $j$  allant de 1 à  $n$ ) **faire**
  - 5:         **si** (*TableVoisinage*[ $j$ ] [ $0$ ] =  $IDN_i$ ) **alors**
  - 6:             *supprimer la ligne  $j$  de la table ;*
  - 7:         **fin** **si**
  - 8:     **fin** **pour**
  - 9:     *Envoyer le paquet *repINST* contenant la nouvelle table vers le nœud successeur ;*
  - 10: **fin** **pour**
-

### Exemple

Cet exemple montre le principe de minimisation de la taille de la table de voisinages transmis par le nœud 4.

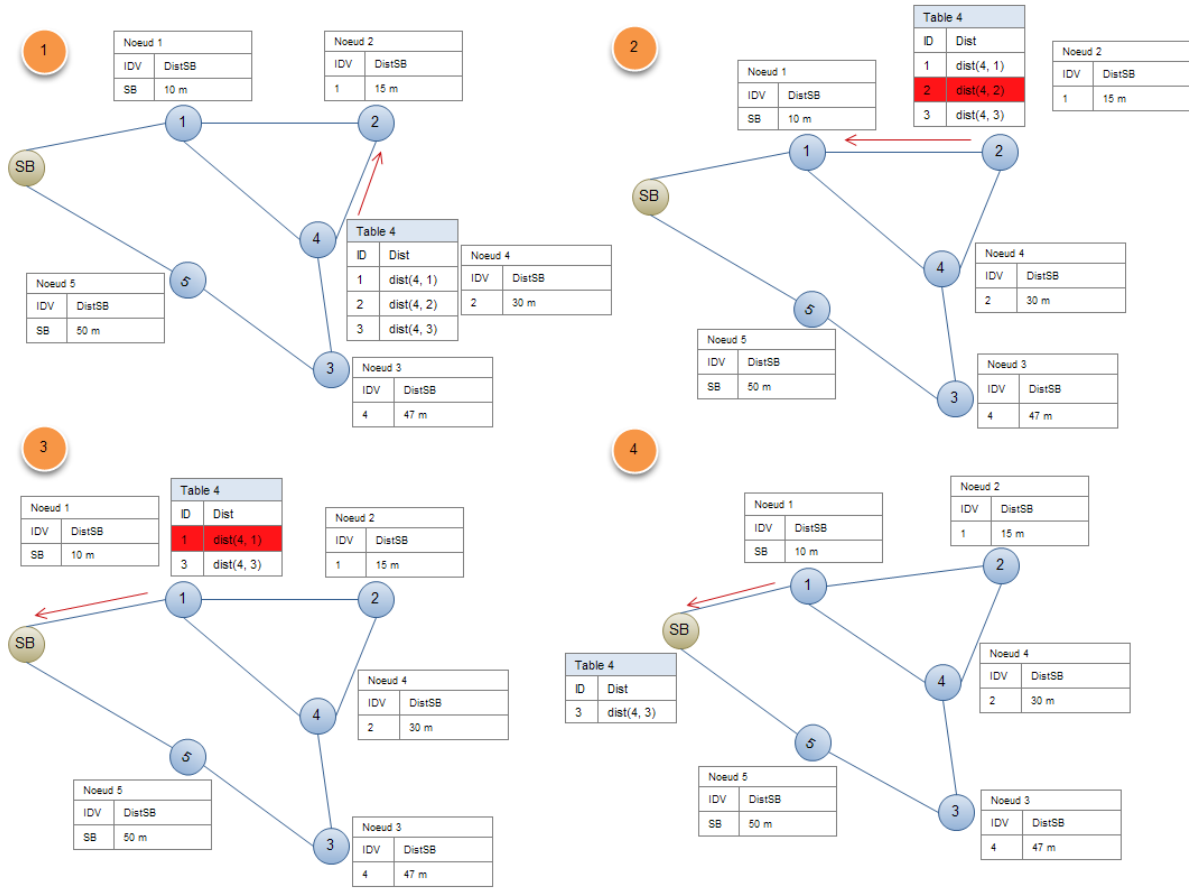


FIGURE III.6 – Exemple de la minimisation des tailles des tables de voisinages.

A la réception de tous les paquets *repINST* contenant les tables de voisinages, la SB est en mesure de construire la matrice de topologie (fig III.6).



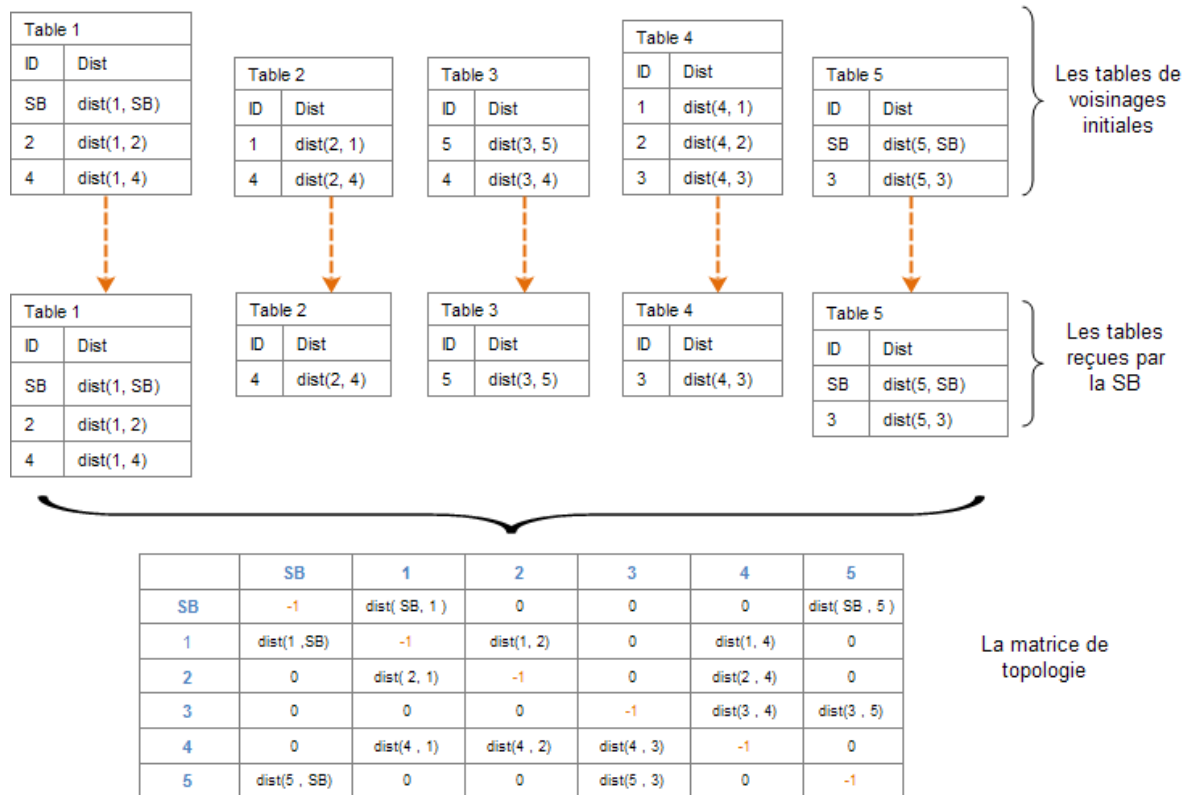


FIGURE III.7 – La construction de la matrice de topologie.

### III.7.2 Phase de transmission

L'objectif de cette phase est de trouver le chemin à multisauts assurant l'équilibrage d'énergie vers la destination, et de minimiser la taille des paquets envoyés. Quand la SB veut acquérir une information à partir d'un capteur précis, elle envoie une requête au capteur en question en lui précisant la route d'aller et retour la plus optimale.

Cette phase est divisée en deux étapes, la première étape, consiste à chercher le chemin optimal parmi tous les chemins menant vers le nœud destination, tout en assurant un équilibrage de l'énergie consommée par chaque nœud constituant le réseau. Elle se fait à chaque fois que la SB veut obtenir des informations auprès d'un nœud précis. La deuxième étape consiste à router les paquets vers la destination, en minimisant leur taille, d'où la réduction de l'énergie de transmission.

#### III.7.2.1 Etape 1 : Choix du chemin optimal

Notre première réflexion pour développer cette partie est la suivante :

- a - La station de base détecte tous les chemins menant au nœud destination en se basant sur la matrice de topologie.

- b - En s'appuyant sur le cout énergétique de transmission des paquets (voir la section 3), la SB estime pour chaque chemin le niveau d'énergie résiduelle de chaque nœud appartenant à ce chemin si ce dernier est élu pour router le paquet vers la destination.
- c - Elle vérifie si l'énergie résiduelle estimée de chaque nœud appartenant à un chemin donné est supérieure à son seuil critique. Si ce n'est pas le cas elle élimine ce chemin de l'ensemble des chemins détecté. En effet, la SB doit vérifier à chaque transmission, si un capteur intermédiaire  $N_i$  puisse être utilisé. Il doit avoir assez d'énergie pour deux émissions, deux réceptions et deux traitements. Par contre, la destination doit avoir assez d'énergie pour une émission, une réception et un traitement.

Le chemin opté par la SB consiste à utiliser les capteurs moins consommateurs d'énergie, c'est à dire, utiliser le chemin  $i$  qui satisfait la fonction suivante :

$$\text{Max} (\text{Min} (\text{niveaux d'énergie de chaque capteur appartenant au chemin } i))$$

Cette dernière est appliquée sur les niveaux d'énergie des capteurs estimés dans l'étape  $b$  qui doivent être supérieures au seuil critique de chaque nœud. Dans le cas où il y'a plus qu'un seul chemin qui possède la même valeur maximale, alors nous comparons les deuxièmes minimum de ces chemins et nous prenons le maximum, et ainsi de suite. Un troisième critère sera utilisé en cas d'égalité consiste à choisir le chemin qui comporte un minimum de capteurs.

- d - Après l'élection d'un chemin vers la destination, la SB mis à jour le vecteur d'énergie en parcourant les nœuds à utiliser pour le routage et en mettant les nouvelles valeurs d'énergie, estimées dans l'étape  $b$ , pour les capteurs concernés(voir Algorithme 3).

Le chemin optimal choisi par la SB est le même chemin emprunté par les paquets pour l'aller et pour le retour.

---

**Algorithm 3** Recherche du chemin optimal

---

- 1: *Détecter tous les chemins menant vers la destination.*
  - 2: **pour** (*chaque chemin  $i$  détecté*) **faire**
  - 3:     *Estimer le niveau d'énergie résiduelle de chaque nœud dans le chemin si ce dernier est élu pour router les paquets ;*
  - 4:     **pour** (*chaque nœud  $N_i$  appartenant au chemin  $i$* ) **faire**
  - 5:         **si** (*Niveau d'énergie résiduelle de  $N_i$  < seuil*) **alors**
  - 6:             *Eliminer le chemin  $i$  de l'ensemble des chemins détectés. ;*
  - 7:         **sinon**
  - 8:             *Enregistré le niveau d'énergie résiduelle minimum du chemin  $i$  ;*
  - 9:         **finsi**
  - 10:     **fin pour**
  - 11: **fin pour**
  - 12: *Sélectionné le chemin dont le niveau d'énergie résiduelle d'un nœud appartenant à ce chemin est le maximum parmi tous les niveaux d'énergie enregistrés ;*
  - 13: *Mettre à jour le vecteur d'énergie ;*
-

Exemple

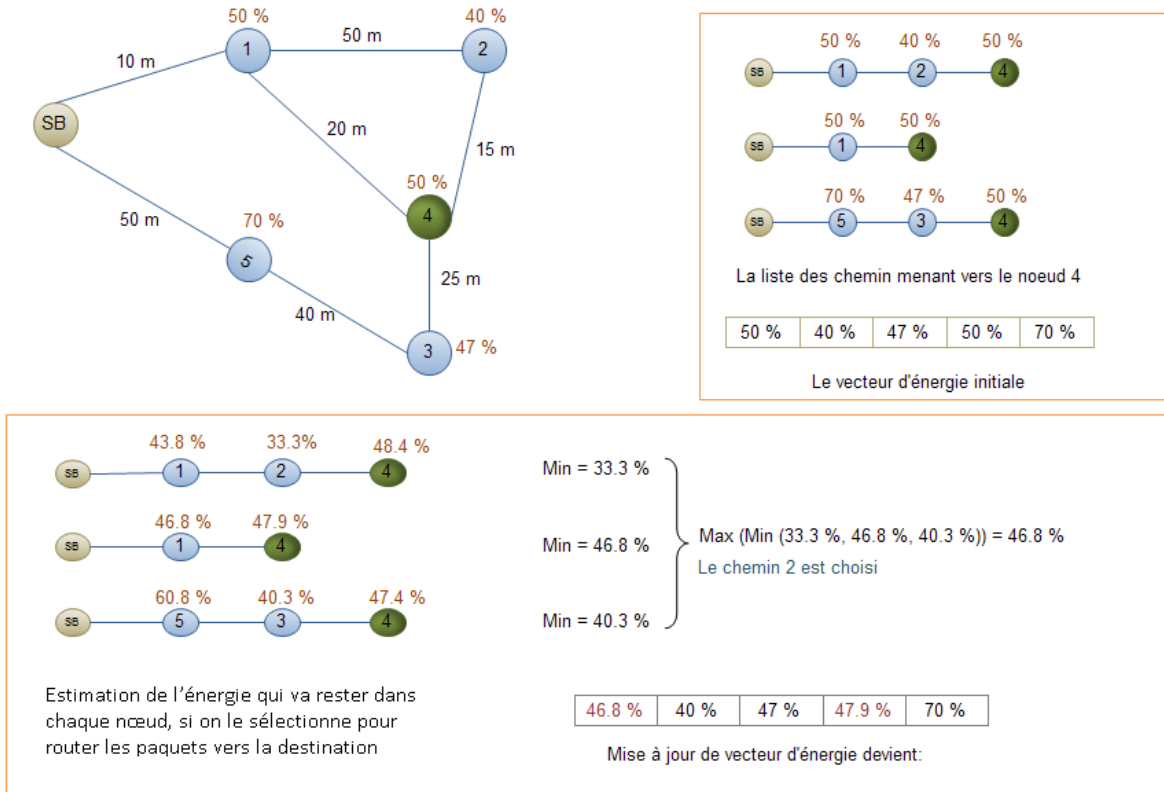


FIGURE III.8 – Exemple d’application de la procédure Max (Min( )).

III.7.2.2 Etape 2 : Routage de paquet

La deuxième étape dans cette phase, consiste à acheminer les paquets vers leurs destinations et à minimiser leur taille. La première solution proposée est la suivante :

- a - La SB construit un paquet *req* contenant l’ID du paquet, le chemin et la requête, puis elle l’envoie au premier nœud dans le chemin.
- b - A l’arrivée du paquet *req* à un nœud intermédiaire  $N_i$  :
  1. Le nœud  $N_i$  mémorise l’identifiant de son prédécesseur conjointement avec l’ID du paquet afin de distinguer entre les paquets qui passe par ce nœud.
  2. Il identifie son successeur en consultant le chemin et il lui envoie ce paquet après avoir éliminé son identifiant du chemin.
  3. Nous répétons les étapes 1, 2 et 3 jusqu’à l’arrivée du paquet à la destination.
- c - Dès que le nœud destination reçoit le paquet *req* :
  1. Il mémorise l’identifiant de son prédécesseur.

2. Il construit un paquet *rep* contenant la réponse à la requête envoyée par la SB (ayant le même identifiant que l'identifiant du paquet *req*).
3. Il envoie ce paquet *rep* au nœud dont il a mémorisé son identifiant durant l'acheminement du paquet *req*.

d - A l'arrivée du paquet *rep* à un nœud intermédiaire  $N_i$  :

1.  $N_i$  vérifie d'abord l'*ID* de *rep*, et il l'envoie au nœud dont son identifiant est mémorisé lors de l'acheminement du paquet *req* ayant le même *ID* que *rep*.
2. Nous répêtons 1 jusqu'à l'arrivée du paquet *rep* à la SB.

En effet, cette solution de base n'est pas optimale, car si le réseau contient  $n$  capteurs et le chemin d'aller et de retour utilisent, au pire des cas tous les nœuds du réseau, nous aurons  $(n - 1) * n / 2$  identifiants à envoyer durant le processus de routage.

### Exemple

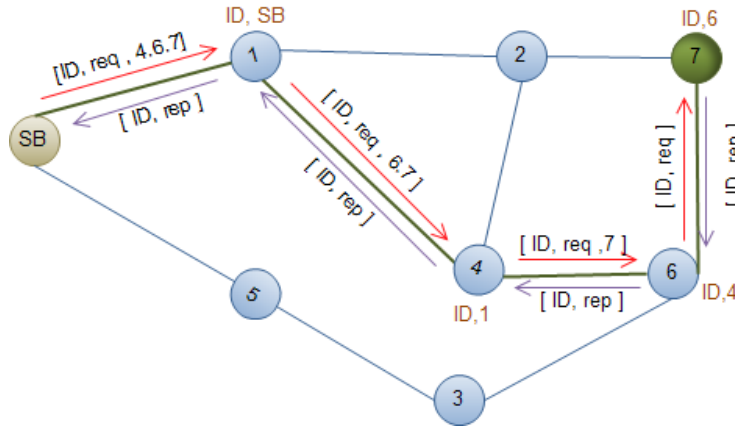


FIGURE III.9 – Exemple de l'acheminement des paquets

Nous proposons dans ce qui suit une nouvelle amélioration qui permet de minimiser la taille des paquets en envoyant une seule valeur entière, ensuite chaque nœud déduit le prochain nœud successeur en appliquant une fonction spécifique sur cette valeur. La solution est énoncée comme suit :

Après la sélection du chemin optimal  $R = A_1 A_2 \dots A_n D$  par la SB. Cette dernière calcule une valeur  $B_0$  en utilisant le modèle mathématique suivant :

$$\begin{cases} B_n = \mathcal{D} \\ B_{n-1} = A_n \times B_n + \mathcal{D} \\ \vdots \\ B_{i-1} = A_i \times B_i + A_{i+1} \\ \vdots \\ B_1 = A_2 \times B_2 + A_3 \\ B_0 = A_1 \times B_1 + A_2 \end{cases}$$

Ensuite, elle envoie  $B_0$  vers le 1<sup>er</sup> nœud du chemin  $R$  (le nœud  $A_1$ ). Lors de la réception de  $B_0$  par le nœud  $A_1$ , il détermine son successeur  $A_2$  et la valeur de  $B_1$  à transmettre pour ce dernier sachant que  $B_0 = A_1 * B_1 + 1 * A_2$ , où  $A_1$  est le nœud courant. La déduction de  $A_2$  et  $B_1$  par le nœud  $A_1$  se fait par la résolution de l'équation diophantienne<sup>1</sup>  $ax + by = c$  tel que  $a = A_1$ ,  $b = 1$  et  $c = B_0$  dont les valeurs  $A_1$ ,  $b$  et  $B_0$  sont connus. Pour résoudre l'équation précédente, nous utilisons l'algorithme d'*Euclide étendu* [18], cela permet de déterminer les valeurs de  $A_2$  et  $B_1$ . Ensuite le nœud  $A_1$  enregistre l'identifiant de son prédécesseur et envoie  $B_1$  vers le voisin dont l'identifiant est  $A_2$ .

Le successeur ré-exécute le même processus pour déterminer le voisin à qui transmettre le paquet  $req$  ainsi que la valeur de  $B_i$  jusqu'à ce que la valeur de  $B_i$  calculé par l'un des nœuds (le nœud  $A_n$ ) soit égale à l'identifiant de son successeur, cela indique que ce dernier est la destination. Le routage du paquet  $req$  suit le chemin inverse (voir Algorithme 4).

---

**Algorithm 4** Routage du paquet  $req$ 

---

```
1: pour (chaque nœud) faire
2:   si (nœud = SB) alors
3:      $n \leftarrow \text{taille}(R)$ ;
4:      $B[n] \leftarrow D$ ;
5:     pour ( $i$  allant de  $n$  à 1) faire
6:        $B[i-1] \leftarrow IDA[i] * B[i] + IDA[i+1]$ ; // IDA contient l'ensemble des identifiant
          // des nœuds appartenant au chemin  $R$ 
7:     fin pour
8:     Envoyer ( $req, B_0$ ) vers le nœud  $A_1$ ;
9:   sinon
10:    pour (chaque voisin  $j$  sauf l'émetteur de nœud  $N_i$ ) faire
11:      Enregistrer l'identifiant de l'émetteur et l'ID du paquet  $req$ ;
12:       $m \leftarrow (B_i - ID_j) \bmod IDN_i$ ;
13:      si ( $m = 0$ ) alors
14:         $B_{i+1} \leftarrow (B_i - ID_j) / IDN_i$ ;
15:        si ( $B_i = ID_j$ ) alors
16:          Envoyer ( $req$ ) vers le nœud  $N_j$ ;
17:        sinon
18:          Insérer  $B_i$  dans le paquet  $req$  puis envoyer le au nœud  $N_j$ ;
19:        finsi
20:      finsi
21:    fin pour
22:  finsi
23: fin pour
```

---

<sup>1</sup>Une équation diophantienne, en mathématiques, est une équation dont les coefficients sont des nombres entiers et dont les solutions recherchées sont également entières.

## Exemple

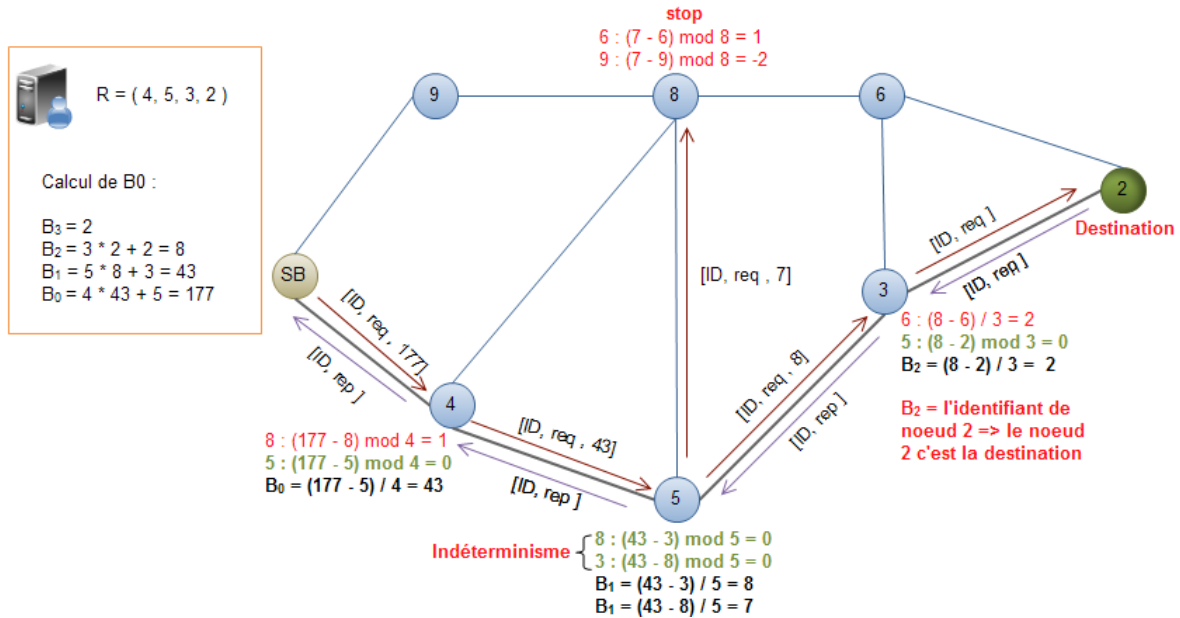


FIGURE III.10 – Exemple d’acheminement des paquets(deuxième solution)

Dans des cas exceptionnels, nous pouvons trouver différents voisins qui possèdent des identifiants qui satisfont l’équation, c’est l’indéterminisme. Dans ce cas, le nœud envoie les différentes valeurs de  $B_i$  calculés à tous les voisins qui satisfont l’équation. Des faux successeurs peuvent être détectés immédiatement dans le prochain saut, et donc dans ce cas, le nœud successeur va arrêter la transmission du paquet req. Nous pouvons réduire la possibilité de l’indéterminisme approximativement à 0 en choisissant des nombre premier pour les identifiants des nœuds.

## III.7.3 Phase de maintenance

La SB doit être en mesure de connaître la situation du réseau à tout moment, d’où la nécessité d’une maintenance de la topologie. De ce fait, si un capteur meurt, la SB doit mettre à jour la matrice de topologie et le vecteur d’énergie.

Notre réflexion vis-à-vis de ce problème est comme suit : la SB calcule le niveau d’énergie de chaque capteur, en se basant sur l’estimation du niveau d’énergie consommé lors des différentes transmissions, et l’estimation de l’énergie consommée lors du captage de données. Cette dernière estimation est faisable vu que les capteurs sont homogènes et ils possèdent des capacités égales (même énergie initiale, même capacité de stockage, même portée du signal et de traitement de données), par conséquent, l’énergie consommée lors du captage de données durant une durée  $t$  est la même pour tous les capteurs. Donc, si un capteur ne peut plus fonctionner en raison de son niveau d’énergie épuisé par rapport au seuil critique,

calculé par la SB, cette dernière met à jour la matrice de topologie et le vecteur d'énergie par l'élimination du nœud concerné.

## III.8 Conclusion

Dans ce chapitre, nous avons détaillé notre solution qu'a comme but la conservation de l'énergie lors de l'acheminement des données. Le gain d'énergie dans notre protocole est dû premièrement au fait que la SB s'en charge d'établir le chemin permettant d'équilibrer la consommation énergétique des capteurs constituant le réseau, afin que ce dernier dure plus longtemps ; et le fait que les capteurs n'ont pas le rôle de choisir les routes, mais juste capter l'information et l'envoyer en suivant le chemin prédéfini. Deuxièmement, nous avons proposé une nouvelle technique qui sert à minimiser la taille des paquets, en permettant à chaque nœud appartenant au chemin menant vers la destination de déduire le prochain nœud successeur à l'aide d'une fonction spécifique.

# Conclusion générale et perspectives

Les réseaux de capteurs constituent un axe de recherche très fertile et peuvent être appliqués dans plusieurs domaines d'applications différents. Ces derniers feront de cette technologie émergente une partie intégrale de nos vies actuelles et futures. L'énergie est la ressource la plus précieuse dans un réseau de capteurs, parce qu'elle influe directement sur la durée de vie des capteurs et du réseau en entier par conséquence. Dans ce mémoire, nous nous sommes intéressés au problème du routage dans les réseaux de capteurs sans fil. Nous avons proposé une solution permettant d'offrir une meilleure prise en compte des ressources énergétiques du réseau.

Dans le premier chapitre de notre travail, nous avons présenté une introduction aux réseaux de capteurs sans fils, où nous avons présenté certaines applications potentielles qui démontrent leur utilité ainsi que les différentes problématiques révélées dans ce type de réseaux et aussi nous avons exposé les causes de consommation d'énergie pour un capteur et pour l'ensemble du réseau. Le deuxième chapitre, nous avons présenté un état de l'art sur les principaux protocoles de routage dans les réseaux de capteurs ainsi que leur classification et nous nous sommes intéressés plus précisément à ceux qui traitent le problème de la consommation d'énergie. Ensuite, dans le troisième chapitre nous avons proposé une solution qui vise à exploiter plus équitablement l'énergie des nœuds constituant le réseau, et à économiser l'énergie dissipée lors de l'acheminement des données.

En matière d'avantages, dans notre protocole, la station de base est sensée trouver les routes les plus optimales pour le routage, et ceci, grâce à la matrice de topologie et le vecteur d'énergie, avec cette façon, les capteurs sont allégés de la fonction du routage. En plus lors de l'acheminement des données notre protocole minimise la taille des paquets du fait que nous avons utilisé une technique permettant à chaque nœud participant dans le processus de routage de déduire le nœud successeur, et ceci, en exploitant une valeur contenant dans le paquet reçu. La minimisation de la taille des paquets implique la minimisation de l'énergie de transmission lors du routage. Dans notre protocole la maintenance de la topologie est assurée par la SB sans la participation des autres nœuds ce qui évite la consommation de leur énergie. De ce fait, si un capteur meurt, la SB met à jour la matrice de topologie et le vecteur d'énergie en estimant l'énergie consommée lors des différentes transmissions et lors de la phase de captage de données.

En guise de perspectives ultérieures à notre travail, nous envisageons en premier lieu, à évaluer la pertinence de notre protocole en finalisant les simulations qui sont en cours.



En deuxième lieu, nous suggérons l'application de notre contribution, plus précisément la technique de minimisation de la taille des paquets de la deuxième étape de la phase de transmission, dans les réseaux ad hoc. Une autre perspective de ce travail, consiste à l'adaptation de notre protocole aux applications orientées événements .

# Bibliographie

- [1] A.Ayache and D.Mouloudj. Routage avec conservation d'énergie dans les réseaux de capteurs sans fils. 2010.
- [2] A.Makhoul. *Réseaux de capteurs : localisation, couverture et fusion de données*. PhD thesis, Laboratoire d'Informatique de l'Université de Franche-Comté (LIFC), 2008.
- [3] A.Manjeshwar and D.P. Agrawal. Teen : A routing protocol for enhanced efficiency in wireless sensor networks. *IEEE*, pages 2009–2015, 2001.
- [4] A.Manjeshwar and D.P.Agrawal. Apteen : A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. *Proceedings of the International Parallel and Distributed Processing Symposium IEEE*, 2002.
- [5] C.Intanagonwiwat, R.Govinda, and D.Estrin. Directed diffusion : A scalable and robust communication paradigm for sensor networks. *Proceeding of the International Conference on Mobile Computing and Networking*, 2000.
- [6] C.Intanagonwiwat, R.Govinda, and D.Estrin. Directed diffusion : A scalable and robust communication paradigm for sensor networks. *Proceeding of the International Conference on Mobile Computing and Networking*, 2000.
- [7] D.Braginsky and D.Estrin. Rumor routing algorithm for sensor networks. *ACM*, September 2002.
- [8] F.Akyiliz, W.Su, Y.Sankarasubramaniam, and E.Cayici. A survey on sensor network. *IEEE Communication Magazine*, pages 0163–6804, 2002.
- [9] G.Chalhoub. Les réseaux de capteurs sans fil. 2010.
- [10] G.Pujolle. *Les réseaux Édition 2008*. EYROLLES, 2008.
- [11] H.Qi, P.T.Kuruganti, and Y.Xu. The development of localized algorithms in wireless sensor networks. July 2002.
- [12] I.F.Akyildiz, W.Su, Y.Sankarasubramaniam, and E.I.Cayirci. Wireless sensor networks : A survey. *IEEE Computer Networks*, pages 393–422, Mars 2002.
- [13] J.Kulik, W.R.Heinzelman, and H.Balakrishnan. Negotiation-based protocols for disseminating information in wireless sensor networks. *ACM MOBICOM*, 1999.

- [14] J.Xu, W.Liu, F.Lang, Y.Zhang, and C.Wang. Distance measurement model based on rssi in wsn. *Jornal of Wireless Sensor Network* 606-611, 2(2010).
- [15] H-C. LE. *Optimisation d'accès au médium et stockage de données distribuées dans les réseaux de capteurs*. PhD thesis, Université de Franche-Comté, 2008.
- [16] L.Jin. *Contribution à l'étude de modèles radio ultra faible consommation pour les réseaux de capteurs en gamme millimétrique*. PhD thesis, Université des sciences et technologies de Lille, 2010.
- [17] L.Khelladi and N.Badache. Les réseaux de capteurs : état de l'art. Février 2004.
- [18] Bishop M. Computer security : Art and science. *Addison Wesley Professional*, 2004.
- [19] M.Ilyas and I.Mahgoub. *Handbook of sensor networks : compact wireless and wired sensing systems*. Crc press LLC, 2005.
- [20] M.Lehsaini. *Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique*. PhD thesis, Université A.B Tlemcen Faculté des Sciences pour l'Ingénieur et l'université de Franche-Comté U.F.R Sciences et Techniques, 2009.
- [21] N.Cuppens-Boulahia and Owezarski P. *Sar ssi 2008*. Editions Publibook, 2008.
- [22] K.Chandrima Rahman. A survey on sensor network. *GCIT*, 2010.
- [23] S.Boulfekhar. *Approches de minimisation d'énergie dans les réseaux de capteurs*. Mémoire de magistère, Université Abderahmane Mira de Bejaia, 2006.
- [24] S.Lindsey and C.Raghavendra. Pegasis :power-efficient gathering in sensor information systems. *IEEE Aerospace Conference Proceedings*, pages Vol. 3, 9–16 pp. 1125–1130, 2002.
- [25] T.He, J.A Stankovic, C.Lu, and T.Abdelzaher. Speed : A stateless protocol for real-time communication in sensor networks. *23rd IEEE International Conference on Distributed Computing Systems (ICDCS'03)*, 2003.
- [26] W.R.Heinzelman, A.Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. *Proceedings of the 33rd Hawaii International Conference on System Sciences IEEE*, 2000.
- [27] Dodge Yadolah and Melfi Giuseppe. *Premiers Pas En Simulation*. Springer-Verlag France, 2008.
- [28] Y.Challal. Réseaux de capteurs sans fils. Novembre 2008.
- [29] Y.Yaser. *Routage pour la gestion de l'énergie dans les réseaux de capteurs sans fil*. PhD thesis, Universite de Haute Alsace Faculté des Sciences et Techniques, 2011.
- [30] Y.Yu, R.Govindan, and D.Estrin. Geographical and energy-aware routing : A recursive data dissemination protocol for wireless sensor networks. 2001.

- [31] A. M. Zungeru, L. Minn Ang, and K. P Seng. Classical and swarm intelligence based routing protocols for wireless sensor networks : A survey and comparison. *Journal of Network and Computer Applications*, pages 1084–8045, Mars 2012.