République algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

Université Mouloud Mammeri de Tizi-Ouzou

Faculté de génie électronique et d'informatique

Département informatique

Mémoire de fin d'étude

Pour l'obstination du diplôme master en informatique

Option : Système Informatique

Thème

Implémentation d''un mécanisme d''agrégation des données dans le protocole de routage LEACH

Réalisé par :

M^elle GAOUA LILIA

Encadré par :

M^me Aoudjit Rachida

Promotion : 2010-2011.

Remerciement

Je tiens à remercier mon Dieu, le tout puissant, de m'avoir donné le courage et la patience jusqu' à l'achèvement de ce travail.

J'exprime ma profonde reconnaissance et mes sincères remerciements à ma promotrice M^me^ Aoudjit Rachida pour sa disponibilité et son encadrement, ainsi que pour son soutien tout au long de l'année.

Je tiens également à remercier les membres du jury pour avoir accepté de juger ce travail.

J'aimerais également remercier tous mes amis et mes collègues de leur soutien et aide et qui m'ont donné la force pour continuer.

Je souhait inclure dans mes remerciements les personnes qui ont bien voulu me faire part de leur expérience dans le domaine.

Mes remerciements vont aussi à tous ceux qui ont contribué de prés ou de loin à la concrétisation de ce travail. Qu'ils trouvent tous ici l'expression de notre gratitude et notre parfaite considération.

Dédicaces

Je dédie ce travail à mes parents, qui m'ont aidé à devenir ce que je suis et pour avoir été là à tous moments.

À mes sacrés frères sœurs ; sarah, toufik et moumouh pour leurs soutiens sans faille.

À tous mes amis et à toutes leurs familles.

lilia

Table des matières

CHAPITRE II LE ROUTAGE DANS LES RCSF

CHAPITRE VI IMPLEMENTATION ET SIMULATION

Liste des figures

Chapitre I :

Chapitre II :

Chapitre III :

Chapitre IV :

Chapitre V :

Chapitre VI :

LISTE DES TABLEAUX

Chapitre I

Chapitre II

Chapitre VI

Introduction générale

A l'heure actuelle, le thème des réseaux de capteurs sans fil (RCSF) provoque un intérêt croissant. Ceci est dû essentiellement aux caractéristiques inhérentes à cette technologie, et qui la favorisent pour un large étendu d'applications dans plusieurs domaines. Parmi ces caractéristiques, on cite la possibilité de déploiement aléatoire du réseau dans des environnements hostiles tels que les champs de bataille, en plus de l'auto-organisation et le fonctionnement autonome des nœuds capteurs.

Cependant, un des verrous majeurs pour la large diffusion des RCSF est celui de l'autonomie en énergie qui nécessite encore beaucoup de travaux de recherche. Il constitue un domaine d'investigation très actif comme le démontre la forte participation internationale à la conférence PowerMEMS en 2008 à Sendai.

De plus, un capteur peut se trouver dans un état où il ne peut pas traiter ou transmette les données. Dans cet état on dit que le capteur en question est congesté, et on parle d'état de congestion. Quand la congestion se présente, elle cause la dégradation des performances d'un réseau avec une perte de données et une consommation inutile de la bande passante. Pour être performant, les protocoles conçus pour les réseaux de capteurs, principalement les protocoles de routage doivent être capables de contrôler la congestion.

Dans un réseau de capteurs sans fil, les capteurs sont souvent densément déployés dans des zones à surveiller. Ainsi, les capteurs proches peuvent capter la même donnée ou les données très proches. En transmettant ces données redondantes aux nœuds relais, elles peuvent être non seulement la source de gaspillage de l'énergie mais aussi dans la plupart des cas elles sont la source de congestion dans le réseau.

Pour répondre à cette problématique, nous proposons une technique d'agrégation des données. Cela consiste à combiner les données redondantes ou fortement corrélées afin de réduire le nombre de messages transmis par les capteurs et donc pouvoir contrôler la congestion et économisé de l'énergie au niveau de la couche réseau. Pour le développement de notre solution va être appliqué sur l'un des protocoles les plus rependu dans les RCSF : le protocole LEACH,

Pour mieux cerner les enjeux de notre étude, nous introduirons dans un premier chapitre les RCSF, leurs caractéristiques, leurs domaines d'applications ainsi que leurs architectures.
Dans le deuxième chapitre, nous aborderons la notion du routage dans les RCSF ; nous donnerons les classifications et les facteurs de la conception des protocoles de routage, en citant quelques exemples de ces protocoles. L'un de ces exemples est le protocole LEACH que nous expliquerons en détail dans troisième chapitre: son architecture de communication, son algorithme, ses caractéristiques. Avant de proposer notre solution et analyse des résultats obtenus par simulation dans le sixième chapitre, nous ferrons un état de l'art sur les techniques de contrôle de congestion pour les réseaux de capteurs dans le quatrième chapitre et sur les techniques d'agrégation des données dans le cinquième chapitre.

I.1. Introduction

A l'heure actuelle, les réseaux sans fil connaissent une très forte expansion. Ils existent depuis des années, mais l'augmentation de la bande passante et la baisse des coûts ont fait exploser leur croissance.

Dans ce chapitre, nous présenterons en premier lieu les réseaux sans fil en général. Nous détaillerons en second lieu leur décomposition en réseaux avec infrastructure (réseaux cellulaires) et sans infrastructure (réseaux Ad Hoc). Nous entamerons par la suite les RCSF en présentant leurs caractéristiques et les types des capteurs qui existent et aborderons les domaines d'applications des RCSF. Pour terminer, nous décrirons l'architecture protocolaire utilisée dans ce type de réseaux.

I.2. Les réseaux sans fil

Le développement rapide dans le domaine de la technologie sans fil, connu par la facilité de déploiement et le coût relativement faible, a permis à un usager muni d'unité portable (Laptops, PDA, Pen Tablet, etc.), d'accéder à l'information indépendamment de la position géographique et du facteur de temps, en lui permettant une libre mobilité, sans l'astreindre à une localisation fixe.

I.2.1. Définition d'un réseau sans fil

Un réseau sans fil (*wireless network*) est, comme son nom l'indique, un réseau dans lequel les terminaux peuvent communiquer sans liaison filaire. Les terminaux du réseau se déplacent librement, tandis que le système doit assurer toutes les fonctionnalités et tous les services d'un réseau classique. **[ABD07]**

La notion de nœuds se restreint sur les extrémités d'une connexion. Elle peut contenir différents terminaux tel que les routeurs, les ordinateurs, les concentrateurs, les commutateurs, etc. **[SEV06]**

La communication sans fil permet une grande flexibilité d'emploi. En particulier la mise en réseau des sites dont le câblage serait trop onéreux à réaliser dans sa totalité, voire même impossible. En effet, la mise en place des réseaux sans fil n'exige pas de lourds aménagements des infrastructures comme c'est le cas dans les réseaux filaires (creusement de tranchées pour acheminer les câbles, équipement des bâtiments en câblage, goulottes et connecteurs, etc.).

Néanmoins, ils présentent des inconvénients étant donné qu'ils sont caractérisés par une faible puissance d'émission et qu'ils n'offrent pas le même niveau de sécurité que les réseaux câblés, vu la nature contraignante de l'environnement sans fil, qui leur impose plusieurs défis que doivent surmonter les unités mobiles.

Les réseaux avec câbles n'ont pas disparu avec l'apparition des réseaux sans fil. Par conséquent, ces deux types de réseaux cohabitent en donnant naissance aux réseaux hybrides.

I.2.2. Caractéristiques des réseaux sans fil

- **Fiabilité :** La propagation des signaux subit des perturbations (microcoupures, erreur de transfert, timeout) dues à l'environnement qui détériore l'information transmise.

- **Débit :** L'une des limitations principales vient de la faiblesse de la bande passante. Ceci est dû au type du média utilisé. On distingue des réseaux utilisant, par exemple, des communications radio qui peuvent atteindre 20 Mbps **[RRA08]** et des communications Bluetooth avec 3Mbps à 10Mbps **[KEL08]**.

- **Sécurité :** Plus qu'elle ne l'est dans les réseaux filaire, la sécurité est d'une importance primordiale dans les réseaux sans fil. Cela est dû à l'absence du câblage dont il résulte la diffusion de l'information facilitant l'interception à distance et la sensibilité au brouillage augmentant les interférences dans le réseau.

- **Topologie dynamique :** Elle change d'une manière fréquente suite à la mobilité continue des nœuds qui forment la topologie du réseau.

I.2.3. Classes des réseaux sans fil

Les réseaux sans fil peuvent être classés selon l'architecture de communication adoptée en deux catégories : les réseaux cellulaires avec infrastructure et les réseaux Ad Hoc sans infrastructure fixe. Plusieurs technologies sont apparentées aux réseaux cellulaires comme : GPS, WiMax, GPRS, etc., et aux réseaux Ad Hoc comme les RCSF. Dans ce qui suit, ces deux classes de réseaux sans fil seront décrites en détail.

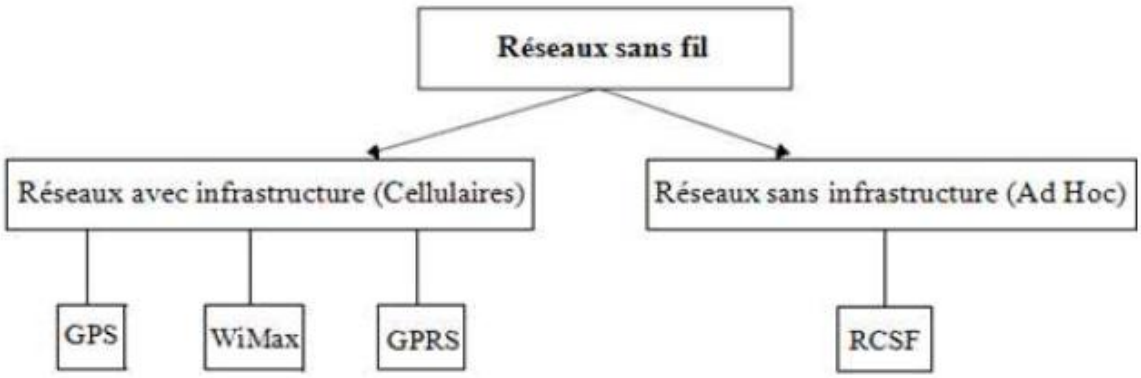


Figure. I-1 : Classes des réseaux sans fil. [LNA04]

I.3. Les réseaux cellulaires

Dans ce mode, le réseau sans fil est composé de deux ensembles d'entités distinctes : les « sites fixes » d'un réseau de communication filaire classique, et les « sites mobiles ». Certains sites fixes, appelés stations de bases (SB) sont munis d'une interface de communication sans fil pour la communication directe avec les sites ou les unités mobiles (UM) localisés dans une zone géographique limitée, appelée cellule. **[TAY00]**

Chaque station de base délimite une cellule à partir de laquelle des unités mobiles peuvent émettre et recevoir des messages. Alors que les sites fixes sont interconnectés entre eux à travers un réseau de communication filaire, généralement fiable et d'un débit élevé. Les liaisons sans fil ont une bande passante limitée qui réduit sévèrement le volume des informations échangées.

Dans ce modèle, une unité mobile ne peut être, à un instant donné, directement connectée qu'à une seule station de base. Elle peut communiquer avec les autres sites à travers la station à laquelle elle est directement rattachée. L'autonomie réduite de sa source d'énergie, lui occasionne de fréquentes déconnexions du réseau; sa reconnexion peut alors se faire dans un environnement nouveau voire dans une nouvelle localisation.

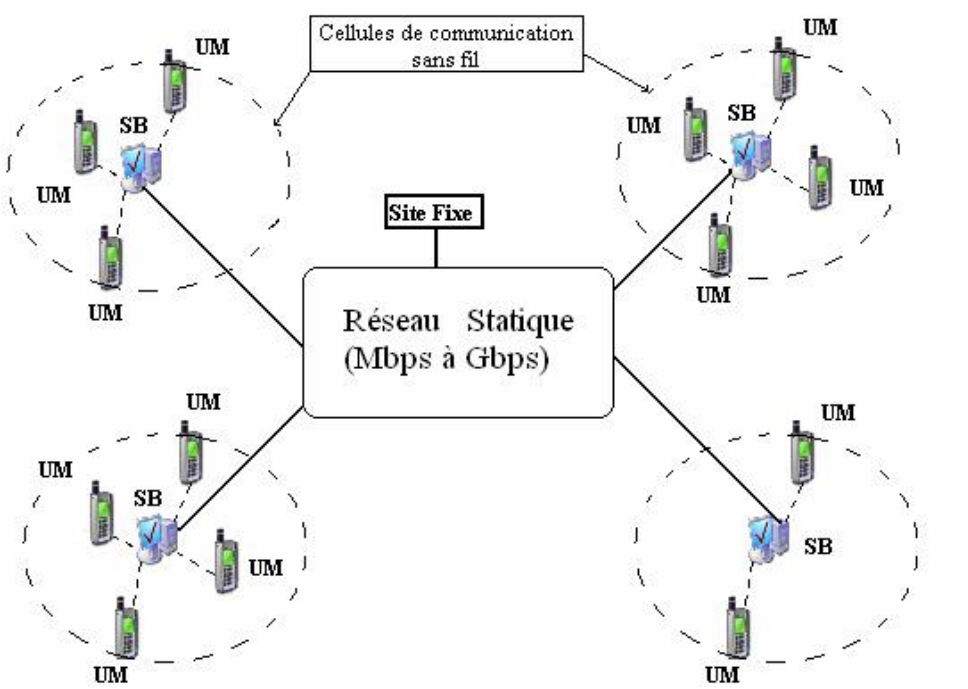


Figure I.2 : *Le réseau cellulaire (GSM)*

A fin d'agrandir la surface de couverture, plusieurs points d'accès1 peuvent être installés pour un même '*groupe de travail'*. Dans le cas d'utilisateurs mobiles, il y'a possibilité de passer d'un point d'accès à un autre sans perte de lien réseau (*comme pour un réseau GSM schématiser dans la figure 2.2*). Cette fonctionnalité s'appelle "Roaming".

I.4. Les réseaux Ad Hoc

À l'heure actuelle, plusieurs systèmes utilisent le modèle cellulaire des réseaux sans fil. Leur inconvénient majeur est qu'ils requièrent une importante infrastructure fixe qui peut être soumise à des risques de destruction dans certaines applications comme le domaine militaire où la capacité à se reconfigurer et à demeurer opérationnelle reste un objectif fondamental. La contrepartie des réseaux cellulaires est les réseaux mobiles Ad Hoc qui sont apparus pour pallier à ce type de désagréments. En effet, les réseaux Ad Hoc ne nécessitent pas une architecture prédéfinie au préalable.

I.4.1. Définition

Les réseaux sans fil ad hoc ou MANET (Mobile Ad Hoc Network) sont composés de terminaux informatiques divers, plus ou moins complexes, appelés nœuds, ayant la possibilité de communiquer de manière autonome par ondes radio en utilisant des interfaces sans fil, sans l'aide d'une infrastructure préexistante ou administration centralisée.

Les nœuds interagissent et peuvent coopérer pour s'échanger des services. Un nœud peut à la fois communiquer directement avec d'autres nœuds ou servir de relais. Un relais permet à des nœuds se trouvant hors de portée radio les uns des autres de communiquer. Ils peuvent exister temporairement pour répondre à un besoin ponctuel de communication d'où la terminologie ad hoc (en latin : pour cela). **[TAY00]**

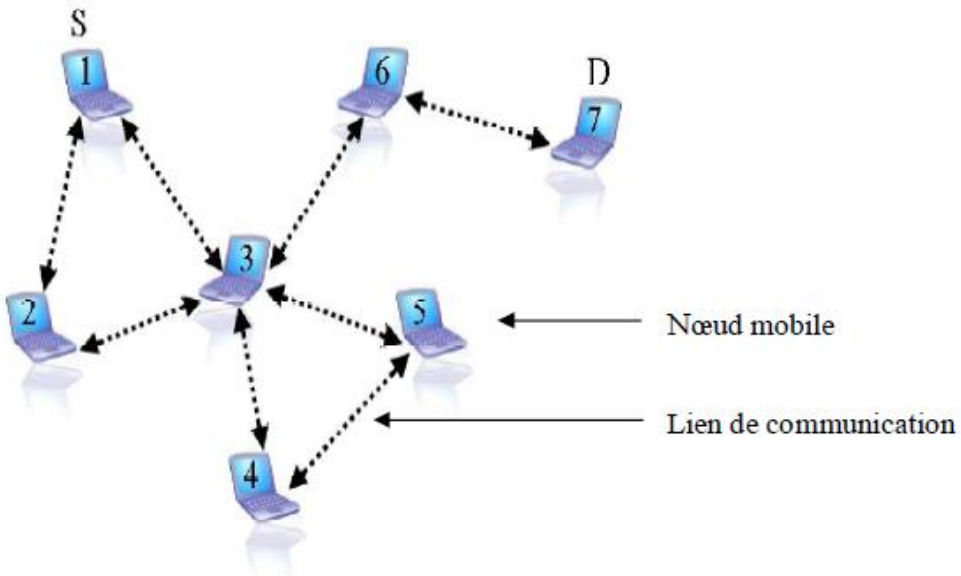


Figure I.3 : un réseau ad hoc.

Si dans le passé, la notion des réseaux Ad Hoc était associée à la communication sur des champs de combat et à l'emplacement des zones dévastées, aujourd'hui, ce n'est plus le cas. En effet, l'utilisation de ce type de réseaux est devenue dans le domaine civil (opérations de secours, incendies, tremblements de terre, missions d'exploration, réseaux de communication, etc.).

I.4.2. Caractéristiques des réseaux Ad Hoc

En plus des caractéristiques des réseaux sans fil en général, les réseaux Ad Hoc ont les caractéristiques suivantes:

- **Architecture décentralisée**: Cela fait référence à un système sans entité centralisée et sans contrôle extérieur. Par conséquent, les nœuds interagissent, analysent et traitent les données sans faire appel à d'autres dispositifs exotiques.

- **Auto-organisation**: Les nœuds découvrent automatiquement et d'une manière autonome les différents paramètres leur permettant de s'intégrer dans l'environnement et de s'auto-configurer pour devenir opérationnels.

- **Sécurité**: L'absence d'infrastructure fixe pénalise l'ensemble du réseau dans la mesure où il faut faire abstraction de toute entité centrale de gestion pour l'accès aux ressources **[BFL03]**. Cela fait que la sécurité dans les réseaux Ad Hoc soit plus pénible à assurer. De plus, les nœuds d'un réseau Ad Hoc assurent la fonction de reconfiguration contrairement à un réseau avec infrastructure où la gestion du rapport de confiance ne se fait qu'entre le nœud et la station. Dans les réseaux Ad Hoc, cette gestion de confiance mutuelle se fait sur tout l'ensemble des nœuds. Par ailleurs, les nœuds Ad Hoc étant fortement mobiles, leur sécurité physique est moins assurée que pour un poste de travail fixe, dans un bureau par exemple. Leur valeur marchande peut être d'une importance non négligeable.

- **La Qualité de Service** (QoS) **:** de nombreuses applications ont besoin de certaines garanties relatives par exemple au débit, au délai, etc. Dans les réseaux ad hoc, ces garanties sont très difficiles à obtenir. Ceci est dû à la nature du canal radio d'une part (interférences et taux d'erreurs élevé) et au fait que des liens entre mobiles peuvent avoir à se partager les ressources (alors qu'en filaire, deux liens sont par définition indépendants). De ce fait, les protocoles de qualité de service habituels ne sont pas utilisables directement dans le monde ad hoc et des solutions spécifiques doivent être proposées.

I.5. Les réseaux de capteurs sans fil

Depuis quelques décennies, le besoin d'observer et de contrôler des environnements hostiles est devenu essentiel pour de nombreuses applications militaires et scientifiques. Les nœuds utilisés doivent être autonomes, d'une taille miniature et peuvent être déployés d'une manière dense et aléatoire dans le champ surveillé. Une classe spéciale des réseaux Ad Hoc appelée réseaux de capteurs sans fil vient au secours. Ceux-ci sont apparus grâce aux développements technologiques tels que la miniaturisation des composants électroniques, la diminution des coûts de fabrication et l'augmentation des performances et des capacités de stockage, d'énergie et de calcul.

I.5.1. Définitions

a) Un capteur

Un capteur est un mini-composant, qui permet d'acquérir des données sur son environnement, les traiter et les communiquer. Son intégration est une tâche difficile à réaliser en tenant compte de certaines contraintes : l'espace mémoire, la consommation énergétique, etc. **[SEV06]**

b) Un réseau de capteurs

Les RCSFs (Réseaux de Capteurs Sans Fil Figure4) sont des types particuliers de réseaux ad hoc. Ils sont composes de dispositifs micro-électroniques appelés : capteurs. Les capteurs sont déployés dans un environnement appelé champs de captage. Les capteurs mesurent une grandeur physique ou chimique. Ils peuvent parfois interagir avec l'environnement extérieur. Les informations recueillies par les capteurs sont envoyées vers un ou plusieurs points de collecte appelés station de base ou puits. La station de base a pour rôle de répondre aux demandes des utilisateurs. Elle peut être fixe ou mobile **[ROU07]**.

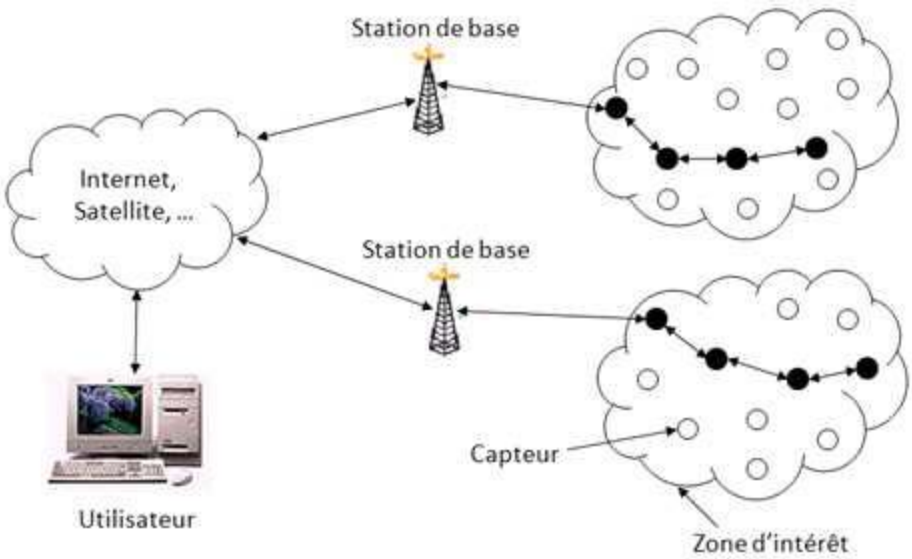


Figure I.4 : un réseau RCSF

I.5.2. Technologies des capteurs

Les recherches dans les RCSF ont débuté par l'agence DARPA pour des besoins de surveillance militaire, avec le projet LWIM et le projet SenseIT qui a été succédé plutard en 1993-1998 par le projet WINS de l'université UCLA en collaboration avec de science Rockwel. Par la suite, d'autres ont vu le jour en 1999-2000 où les participants étaient essentiellement parmi les milieux universitaires.

Par exemple en 1999, UC Berkeley, l'USC, et, MIT avec le projet µAMPS **[BHA08, AHR07]**. Ces projets ont permis le développement de plusieurs types de capteurs:

Compagnie	Université, agence	Caractéristiques	Applications	Capteur
Smart Dust	UC Berkeley	-Small microcontroller: 8 kb code, 512 B données - low-power radio 10 kb -EEPROM storage (32 KB)	-Surveillance -Acquisition de données	 UCB WeC 99"Smart Rock" [ISA07a]
Crossbow	UC Berkeley	-RAM: 4KOctets -ROM/Flash:128kOctets - Flash externe: 512KB -Processeur: Atmel AVR Atmega 128 8-bit 8MHz	-Environnement -Militaire -Sécurité	 MICA2 868 [CRO08]
WINS	UCLA, Rockwell	-microcontroller: 32KB, RAM 1MB bootable flash -StrongARM SA 1100, 32 bit RISC processor, 1MB SRAM, 4MB flash	-Surveillance -Contrôle -Sécurité	 Rockwell : WINS [DJP03]

Tab. I-1 : Technologies des capteurs. [CRO08, ISA07a, DJP03, ADJO9]

1 .5.3. Composant d'un nœud capteur

Un capteur est composé de trois unités alimentées par une source d'énergie embarquée. Ces unités sont illustrées dans la figure suivante.

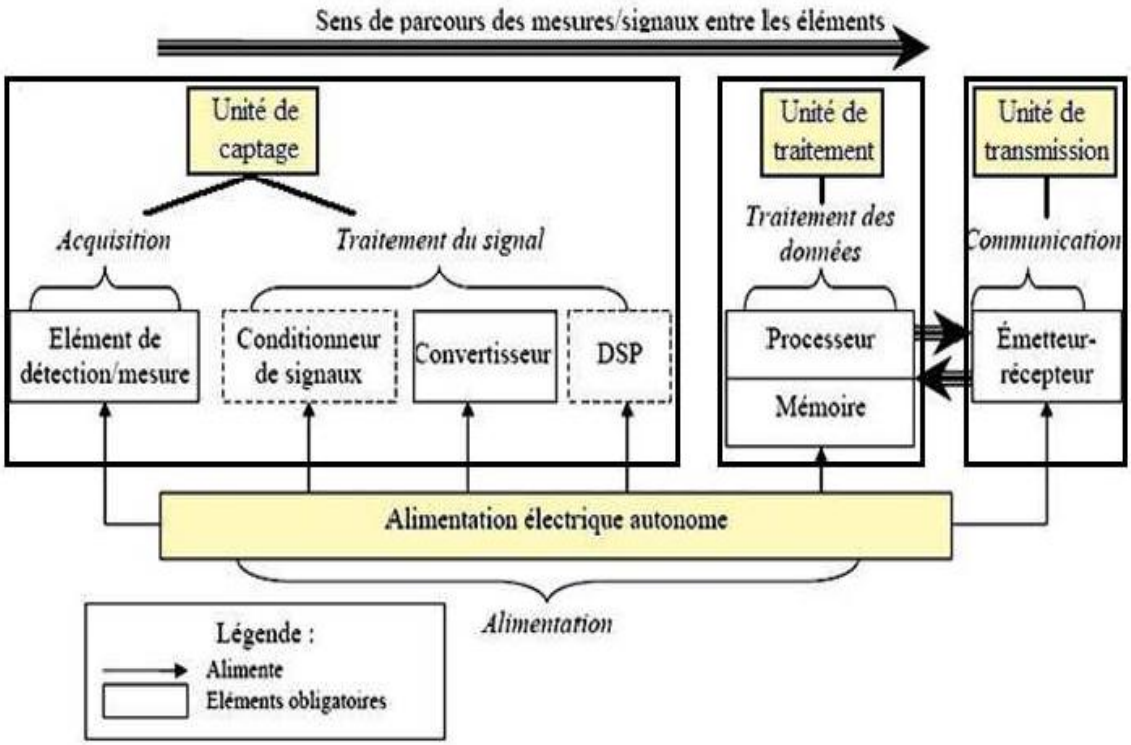


Fig. I-5: Architecture d'un nœud capteur. [SEV06]

- **Unité de captage** :

  Elle est composée de deux sous-unités : une unité d'acquisition, qui permet de détecter les mesures désirées par un capteur, et d'une unité de traitement des signaux qui transforme les mesures analogiques détectées en signaux numériques par un convertisseur analogique numérique. Comme l'indique la figure I-5, la sous-unité de traitement des signaux utilise éventuellement un conditionneur des signaux et un processeur de signal numérique DSP *(Digital Signal Processor).*

- **Unité de traitement:**

  Elle est composée d'une mémoire (unité de stockage) et d'un processeur (unité de calcul) qui permet d'effectuer des calculs simples pour que ce nœud puisse collaborer avec les autres nœuds du réseau. De plus, elle possède deux interfaces :

  - La première, liée avec l'unité de captage par laquelle, elle reçoit les mesures détectées.
  - La seconde, liée avec l'unité de transmission par laquelle, elle communique les données qu'elle a traitées.

- **Unité de transmission** :

  Elle est responsable de toutes les émissions et réceptions de données qui représentent l'état actif du nœud. Par ailleurs, les nœuds peuvent se mettre en veille ou écouter seulement le trafic. L'unité de transmission est l'unité qui consomme le plus d'énergie par apport aux précédentes unités.

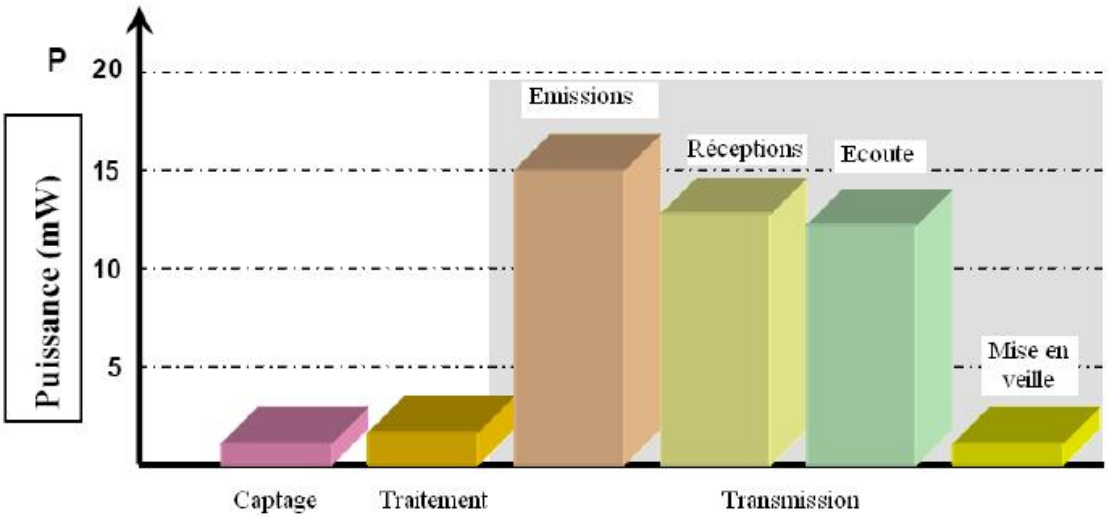


Fig. I-6 : Consommation d'énergie en captage, traitement et transmission. [ATA07]

I.5.4 Topologies des réseaux de capteurs sans fils

Il existe deux types d'architectures pour les réseaux de capteurs : les réseaux de capteurs plats et les réseaux de capteurs hiérarchiques. **[DUT07] [KDT07]**

I.5.4.a Topologie Hiérarchique

Un réseau de capteurs sans fil plat est un réseau homogène, où tous les nœuds sont identiques en termes de batterie et des fonctions, excepté le *« Sink ».* Ce dernier joue le rôle d'une passerelle et est responsable de la transmission de l'information collectée à l'utilisateur final (voir figure II.2).

I.5.4.b Topologie plate (Flat)

Une architecture hiérarchique a été proposée pour réduire le coût et la complexité de la plus part des nœuds capteurs. Elle consiste à introduire un ensemble de nœuds plus coûteux et plus puissants, en créant une infrastructure qui décharge la majorité des nœuds simples à faible coût de plusieurs fonctions du réseau. L'architecture hiérarchique est composée de plusieurs couches : une couche de capteur, une couche de transmission et une couche de point d'accès (voir figure II.3).

1.5.5 Caractéristiques des réseaux de capteurs

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

1.5.5.1 Densité « importante » des nœuds

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

1.5.5.2 Topologie dynamique

La topologie des réseaux de capteurs instable est le résultat des trois facteurs essentiels suivants:

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

1.5.5.3 Auto organisation

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

1.5.5.4 La tolérance de fautes

Le réseau doit être capable de maintenir ses fonctionnalités sans interruptions en cas de défaillance d'un ou plusieurs de ses capteurs. Cette défaillance peut être causée par une perte d'énergie, ou par dommage physique ou interférence de l'environnement. Le degré de tolérance dépend du degré de criticité de l'application et des données échangées [4].

1.5.5.5 Scalabilité

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

I.5.5.6 Contraintes et facteurs conceptuelles des réseaux de capteurs

L'architecture des réseaux de capteurs peut être influencée par certains facteurs pouvant agir sur son bon fonctionnement :

I .5.6.1 Pannes fréquentes

Pour que les paquets de données atteignent la destination finale *« Sink »*, ils vont passer par un grand nombre de nœuds. À cause de sa topologie dynamique et la probabilité de rencontrer un nœud capteur en panne ou mort, le réseau de capteurs ne peut plus garantir des taux de livraison importants. De plus, la communication radio peut être bruitée, ce qui peut endommager l'information transmise.

I.5.6.2 L'environnement

Les nœuds capteurs doivent être conçus d'une manière à résister aux différentes et sévères conditions de l'environnement : forte chaleur, pluie,...

I.5.6.3 La 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.

I.5.6.4 La consommation d'énergie

Un capteur, de part sa taille, est limitée en énergie. Dans la plupart des cas, le remplacement de la batterie est impossible, ce qui veut dire que la durée de vie d'un capteur dépend grandement de la durée de vie de sa batterie. Les recherches actuelles se concentrent principalement sur les moyens d'optimiser la consommation d'énergie par les nœuds capteurs.

I.5.7 Les modèles de transmission des données dans les réseaux de capteurs

La transmission de données dans les réseaux de capteurs peut se faire suivant plusieurs modèles dont on distingue trois essentielles :

- Modèle *event driven*
- Modèle *query driven*
- Modèle *continu*ous

I.5.7.1 Le modèle *driven event*

La génération et la transmission des paquets DATA est commandé par la réalisation d'un événement. La plupart des applications *driven event* **[KDT07]** sont des applications intolérantes aux délais (temps réel), critiques, interactives et de non bout en bout. En fait, au lieu d'avoir un nœud émetteur et un autre récepteur de l'information, on trouve un nœud récepteur (le nœud de contrôle « *sink* ») et un groupe de nœuds capteurs, se trouvant proche de l'événement, qui sont tous des émetteur de la même information. La réussite de ces applications, pour ce modèle, repose essentiellement sur la détection de l'événement et la rapidité des prises des réactions nécessaires pour assurer l'aspect temps réel des applications.

L'inconvénient majeur de ce modèle est la redondance des données. En fait, les nœuds excités par le même événement envoient la même information au nœud de contrôle « *sink* ». Pour cela, un protocole de routage basé sur la négociation des données est recommandé.

I.5.7.2 Le modèle query driven

Le modèle *query driven* est semblable au modèle *driven event* sauf que la collecte des informations sur l'état de l'environnement est initiée par des interrogations envoyées par le nœud de contrôle « *sink* », alors que, pour le modèle précédant, elle est déclenchée suite à un événement détecté. La plupart des applications *query driven* sont des applications interactives, critiques, de non bout en bout et leur tolérance aux délais dépend de l'urgence de l'interrogation **[KDT07]**.

Notons que le modèle *query driven* peut être utilisé pour contrôler et reconfigurer les nœuds. Par exemple, le « *sink* » peut envoyer des commandes au lieu des interrogations pour modifier le programme d'un nœud capteur, modifier son taux de trafic ou son rôle. Seul le nœud capteur jouant le rôle de « *sink* » est autorisé d'émettre des demandes d'interrogations ou des commandes et ce pour assurer l'ordre et l'hiérarchie de réseau de capteur.

I.5.7.3. Le modèle *continuous*

Dans le modèle continu, les nœuds capteurs envoient les informations d'une manière continue au nœud « *sink* » suivant un volume de trafic prédéterminé.

I.5.8. Applications des réseaux de capteurs

Les capteurs sont devenus des éléments incontournables dans tous les systèmes où les informations issues de l'environnement extérieur sont nécessaires pour évaluer et agir. Les RCSF couvrent un large panel d'applications. Ceci grâce aux multiples domaines dans lesquels ils peuvent apporter une performance irréprochable. Les besoins de contrôler, prévoir, observer des phénomènes physiques (mesurer des conditions ambiantes tels que la température, l'humidité, le mouvement de véhicules, les incendies, etc.) ont permis aux RCSF d'envahir de nombreuses applications militaires, industrielles, scientifiques, etc. Dans ce qui suit, des exemples d'applications potentielles sont classifiés en catégories :

I.5.8.1. Applications militaires

Comme dans le cas de plusieurs technologies, le domaine militaire a été un moteur initial pour le développement des RCSF qui permettent la détection et la collection d'informations sur la position de l'ennemi, la surveillance des zones hostiles (contaminées) et la détection d'agents chimiques et bactériologiques dans l'air.

I.5.8.2. Applications à la sécurité

Les altérations dans la structure d'un bâtiment, suite à un séisme ou à un vieillissement, peuvent être détectées par des capteurs intégrés dans les murs ou dans le béton. Un RCSF de mouvements peut constituer un système d'alarme distribué qui sert à détecter les intrusions sur un large secteur.

I.5.8.3. Applications environnementales

Des thermo-capteurs dispersés à partir d'un avion sur une forêt peuvent signaler un éventuel début d'incendie, contrôler la qualité de l'air et recueillir des informations diverses sur l'état du milieu naturel. Sur les sites industriels, les centrales nucléaires ou dans les pétroliers, des capteurs peuvent être déployés pour détecter des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole).

Des RCSF peuvent détecter la présence humaine. Ainsi, la climatisation peut être déclenchée seulement aux endroits où il y a des personnes présentes. Une telle application permet de réduire la demande mondiale en énergie réduisant du même coup l'émission des gaz à effet de serre. Rien que pour les États-Unis, on estime cette économie à 55 milliards de dollars par an avec une diminution de 35 millions de tonnes des émissions de carbone dans l'air. **[TEC08]**

I.5.8.4. Applications médicales

Cette technologie est de plus en plus utilisée dans le domaine médical en se propageant dans de nombreuses applications : la mesure et l'analyse non intrusives de données physiologiques, la surveillance de la température, la fréquence cardiaque, l'oxygénation du sang et le pouls du patient. **[MWI06]**

I.5.8.5 Projets d'applications en cours

Après avoir exposé un certain nombre d'applications potentielles envisagées pour les RCSF, nous décrivons dans ce qui suit des projets d'applications qui sont en cours de réalisation.

- **Newtrax pour les applications militaires**

  Afin de répondre aux exigences des opérations militaires, les forces canadiennes choisissent la compagnie privée Newtrax (qui a comme vision de fournir des solutions de mesure, contrôle, messagerie et localisation) comme partenaire industriel pour un contrat évalué à 1,5 milliards de dollars pour fournir sa technologie des RCSF maillés. Cette architecture offre une résilience et extensibilité supérieure.

De plus, chaque nœud peut être ajusté pour fonctionner selon différentes fréquences entre 100 MHz et 1 GHz **[ATA07]**.

Fig. I-7 : Un service militaire utilisant les RCSF. [ATA07]

- **Glacsweb pour les applications environnementales.**

Glacsweb est un projet réalisé par l'université de Southampton en Grande-Bretagne **[JKR04]**. Il vise à comprendre les changements climatiques et leurs effets sur le niveau de la mer, étude des glaciers. Ces différentes applications permettent d'ores et déjà d'étudier et d'apporter des solutions aux différents problèmes liés à la gestion de réseaux de capteurs de grande envergure

Fig. I-8 : Exemple d'utilisation Glacsweb. [ISA07a]

- **Projet en cours dans le domaine médicale**

Actuellement, des micro-caméras qui peuvent être avalées existent. Elles sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain avec une autonomie de 24 heures. Le projet actuel est de créer une rétine artificielle composée de 100 micro-capteurs pour corriger la vue. Un émetteur-récepteur transmet les données à une station de base branchée à un ordinateur personnel, à un poste des soins infirmiers ou à un assistant numérique. **[JKR04]**

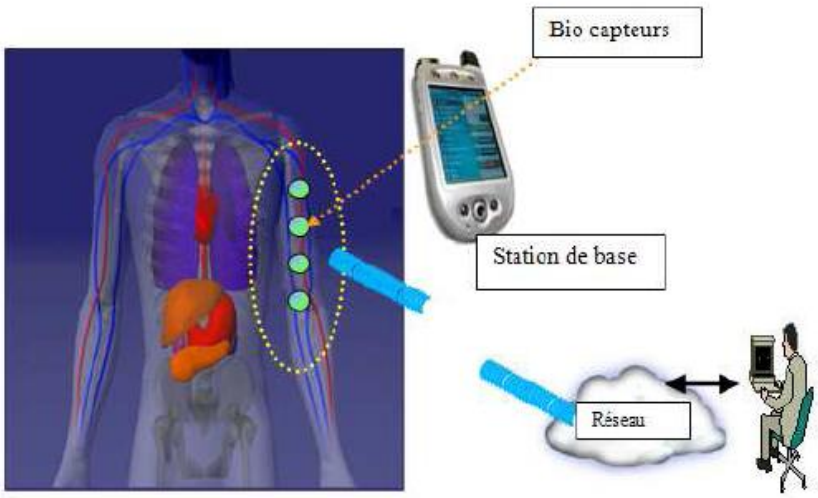


Fig. I-9: Application des RCSF en médecine. [DJA08]

I.5.9 Pile protocolaire des réseaux de capteurs

Dans le but d'un établissement efficace d'un RCSF, une architecture en couches est adoptée afin d'améliorer la robustesse du réseau. Une pile protocolaire de cinq couches est donc utilisée par les nœuds du réseau. Citons la couche application, la couche transport, la couche réseau, la couche liaison de données et la couche physique.

De plus, cette pile possède trois plans (niveaux) de gestion : le plan de gestion des tâches qui permet de bien affecter les tâches aux nœuds capteurs, le plan de gestion de mobilité qui permet de garder une image sur la localisation des nœuds pendant la phase de routage, et, le plan de gestion de l'énergie qui permet de conserver le maximum d'énergie.

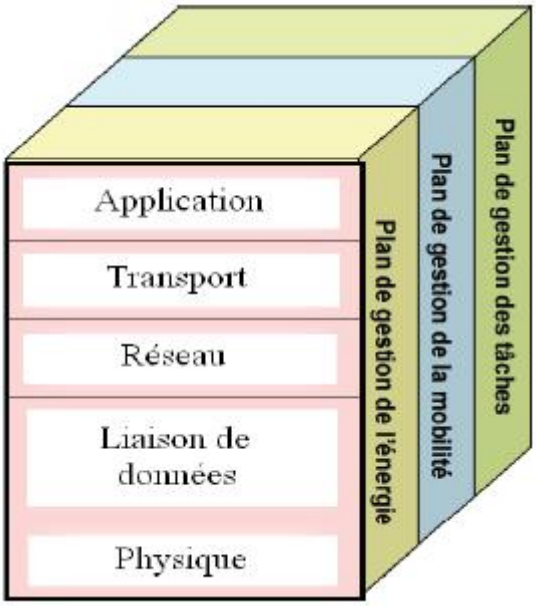


Fig. I-10 : La pile protocolaire dans les réseaux de capteurs. [BOU07]

- **Couche application :** Elle assure l'interface avec les applications. Il s'agit donc de la couche la plus proche des utilisateurs, gérée directement par les logiciels. Parmi les protocoles d'application, nous citons : SMP *(Sensor Management Protocol)* et TADAP *(Task Assignement and Data Advertisement Protocol)*.

- **Couche transport :** Elle vérifie le bon acheminement des données et la qualité de la transmission. Dans les RCSF, la fiabilité de transmission n'est pas majeure. Ainsi, les erreurs et les pertes sont tolérées. Par conséquent, un protocole de transport proche du protocole UDP et appelé UDP-Like *(User Datagram Protocol Like)* est utilisé. Cependant, comme le protocole de transport universel est TCP *(Transmission Control Protocol*), les RCSF doivent donc posséder, lors d'une communication avec un réseau externe, une interface TCP-spliting pour vérifier la compatibilité entre ces deux réseaux communicants.

- **Couche réseau :** Elle s'occupe du routage de données fournies par la couche transport. Elle établit les routes entre les nœuds capteurs et le nœud puits et sélectionne le meilleur chemin en termes d'énergie, délai de transmission, débit, etc. Les protocoles de routage conçus pour les RCSF sont différents de ceux conçus pour les réseaux Ad Hoc puisque les RCSF sont différents selon plusieurs critères comme :

- l'absence d'adressage fixe des nouds tout en utilisant un adressage basé-attribut.
- l'établissement des communications multi-sauts.
- l'établissement des routes liant plusieurs sources en une seule destination pour agréger des données similaires, etc.

Parmi ces protocoles, nous citons : LEACH *(Low-Energy Adaptive Clustering Hierarchy)* et SAR (*Sequential Assignement Routing*).

- **Couche liaison de données :** Elle est responsable de l'accès au media physique et la détection et la correction d'erreurs intervenues sur la couche physique. De plus, elle établit une communication saut-par-saut entre les noeuds. C'est-à-dire, elle détermine les liens de communication entre eux dans une distance d'un seul saut. Parmi les protocoles de liaison de données, nous citons: SMACS *(Self-organizing Medium Access Control for Sensor networks)* et EAR *(Eavesdrop And Register).*

- **Couche physique :** Elle permet de moduler les données et les acheminer dans le media physique tout en choisissant les bonnes fréquences.

I.6. Conclusion

Dans ce chapitre, nous avons présenté les réseaux sans fil en général. Ils sont généralement décomposés selon deux modes : réseaux avec infrastructure ou centralisés (cellulaires) et réseaux sans infrastructure ou décentralisés (Ad Hoc).

Nous avons décrit ensuite les RCSF qui sont apparentés aux réseaux Ad Hoc. Ces réseaux connaissent un grand essor grâce à la multitude d'applications qu'ils offrent ainsi que leurs caractéristiques inhérentes telles que leur déploiement aléatoire et de faible coût, leur grande mobilité et aussi, grâce aux récents développements concernant la miniaturisation des composants électroniques (construction des capteurs de quelques millimètres cubes de volume).

Dans le domaine de la recherche, les RCSF posent un certain nombre de challenges au niveau de la taille des capteurs. Autrement dit, on recherche une miniaturisation maximale et des performances optimales quant à la transmission, le débit et la consommation d'énergie qui est un facteur à prendre en considération dans la conception des RCSF. En effet, la plus grande partie de cette ressource est utilisée pendant la communication. Toutes ces recherches vont dans le même sens: améliorer au maximum les performances d'un RCSF. En revanche chaque tentative d'amélioration d'un critère pose des problèmes majeurs sur un autre. Par exemple la miniaturisation de la batterie pose le problème d'une durée de vie plus courte.
Le routage de données dans un RCSF représente un point très important que nous introduirons dans le chapitre suivant.

II.1 Introduction

En général, le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion donné. Le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau, suivant certains critères de performance.

Dans les RCSF, l'utilisation des protocoles de routage conçus pour les réseaux Ad Hoc traditionnels est inappropriée. Ceci est en raison des caractéristiques par lesquelles se distinguent les deux types de réseaux, d'où la nécessité de les améliorer ou de développer de nouveaux protocoles de routage spécifiques aux RCSF.

Dans ce chapitre, nous commencerons par citer quelques facteurs qui influencent sur la conception des protocoles de routage au sein des RCSF. En second lieu, nous étudierons les métriques qui permettent de tester l'efficacité du protocole une fois conçu. Par la suite, nous décrirons les types de classifications des protocoles de routage dont plusieurs sont de plus en plus perfectionnés.

II.2 Routage et réseaux de capteurs

Le routage dans les réseaux de capteurs consiste à déterminer les routes reliant les nœuds capteurs et le nœud puits et à acheminer les données captées à travers ces routes. Les nœuds d'un réseau de capteurs (ou plus généralement d'un réseau ad hoc) sont équipés d'une interface de communication sans fil à portée limitée. La coopération des nœuds est donc nécessaire pour assurer le service de routage. Les nœuds doivent alors, en plus de leur rôle dans l'acquisition des données, jouer le rôle de routeurs, relayant de la sorte les paquets vers leur destination finale (le nœud puits). Ainsi, une donnée captée peut passer par plusieurs nœuds avant d'atteindre le nœud puits (routage multi-sauts).

Le routage doit être effectué par un algorithme distribué exécuté au niveau de tous les nœuds du réseau. Ceci peut être réalisé soit de manière proactive, où les routes sont maintenues avant d'en avoir besoin, soit de manière réactive où les routes sont établies à la demande.

Le choix de la route reliant une source de données au nœud puits se fait tout en optimisant une certaine métrique (consommation d'énergie, délais de transmission, nombre de sauts,…).

Plusieurs contraintes rendent le routage difficile. Par exemple, un protocole de routage doit maximiser la durée de vie du réseau, ce qui représente une contrainte primordiale. D'autres facteurs doivent être pris en compte et seront abordés dans la section suivante.

II.3 Facteurs de conception de protocoles de routage

La conception des protocoles de routage dans les réseaux de capteurs nécessite la prise en compte de plusieurs facteurs. Ces derniers sont d'une grande importance, car ils servent de directives guidant les concepteurs pour qu'une communication efficace puisse être assurée. En outre, ils peuvent servir de métriques de comparaison entre les différents protocoles de routage proposés pour les réseaux de capteurs **[AKY02]**.

Dans ce qui suit, nous exposerons les principaux facteurs qui influencent la conception des protocoles de routages dans les réseaux de capteurs.

II.3.1 Tolérance aux pannes

La propriété de tolérance aux pannes est définie par l'aptitude du protocole de routage à maintenir ses fonctionnalités, en cas de panne de quelques nœuds. Le but de la tolérance aux pannes est d'éviter la faille totale du système malgré la présence de fautes dans un sous ensemble de ses composants élémentaires **[FAT07]**. Les pannes sont tolérées puisqu'elles sont plus fréquentes à cause de l'épuisement rapide d'énergie et à la nature contraignante de l'environnement qui expose les nœuds à des endommagements physiques. Pour cela, les protocoles de routage conçus doivent atteindre le niveau de tolérance aux pannes requis selon l'application. A cet effet, les protocoles de routage doivent, en cas de défaillance de liens de communication, procéder à la formation de nouvelles routes entre les nœuds.

II.3.2 Consommation d'énergie

L'énergie disponible au niveau des nœuds capteurs est utilisée pour le captage, le traitement des données captées et leur transmission vers le nœud puits. Des études récentes concernant les réseaux de capteurs ont montré que la phase de transmission est la plus consommatrice en énergie. En effet, l'exemple donné dans **[POT00]** montre que la transmission de 1Kb de données sur une distance de 100 mètres consomme une énergie de 3 Joules. Tandis qu'un processeur avec une modeste capacité de calcul de 100 MIPS (Million of Instructions Per Second) peut exécuter 300 millions d'instructions avec la même quantité d'énergie. Cet exemple illustre qu'il serait plus judicieux pour un capteur, au moment du routage, de faire un traitement local et de ne transmettre que les informations utiles, plutôt que d'envoyer toutes les données captées dans leur état brut.

D'autre part, la taille réduite des nœuds capteurs impose l'utilisation de ressources d'énergie limitées telles que les batteries. La durée de vie d'un micro-capteur dépend de la durée de vie de sa batterie, car celle-ci est généralement irremplaçable et est dépourvue d'une unité de rechargement. De plus, les nœuds d'un réseau de capteurs jouent un rôle important dans le maintien de la connectivité du réseau. De ce fait, la durée de vie du réseau de capteurs en entier dépend fortement de celle des nœuds capteurs.

L'efficacité en consommation d'énergie représente alors une métrique de performance significative, dans les réseaux de capteurs, qui influence directement la durée de vie du réseau en entier. Pour cela, les concepteurs peuvent au moment du développement des protocoles, négliger les autres métriques de performance telles que les délais de transmission des données captées, au détriment du facteur de consommation d'énergie **[AKY02]**.

II.2.3 Limitations de capacités des nœuds

Les nœuds ont des capacités de calcul, de stockage et de communication limitées. Les concepteurs de protocoles de routage doivent englober des opérations simples et peu exigeantes en capacité de calcul et de stockage.

II.3.4 La scalabilité

Certaines applications des réseaux de capteurs nécessitent un déploiement d'un grand nombre de nœuds qui peut atteindre des milliers de capteurs. De plus, elles peuvent nécessiter une densité élevée des nœuds-capteurs dans l'environnement de déploiement. Cette densité peut varier, selon l'application, de quelques capteurs jusqu'à plusieurs centaines de capteurs dans une région de taille inférieure à 10 mètres de diamètre. Par exemple, une application de diagnostique des machines industrielles peut nécessiter une densité allant jusqu'à 300 nœuds dans une région de 25 m² **[AKY02]**.

Suivant [BUL01], la densité peut être calculée comme suit :

$$\mu(r) = \frac{(N.p.R^2)}{A}$$

Où :

- A est la surface de la région de captage.
- N est le nombre de nœuds capteurs déployés dans la région A.
- R le domaine de transmission d'un nœud.

μ(r) donne alors le nombre de nœuds se trouvant dans le domaine de transmission R de chaque nœud de la région A.

Les nouveaux protocoles de routage doivent garantir un bon fonctionnement avec ce nombre élevé de capteurs. Ils doivent aussi exploiter la nature fortement dense des réseaux de capteurs.

II.3.5 Connectivité

Le nombre important de nœuds dans un RCSF, fait qu'ils sont généralement dispersés de façon aléatoire, et ne sont pas uniformément répartis sur le champ de captage. Ce qui implique que certaines régions du champ de déploiement puissent bénéficier d'une meilleure connectivité par rapport à d'autres et les phénomènes captés dans ces régions peuvent êtres routés plus facilement. De plus, certains nœuds capteurs peuvent tomber en panne, pendant le fonctionnement du réseau. Il est donc souvent nécessaire de déployer des nœuds supplémentaires pour combler les "trous" et maintenir la connectivité du réseau, ce qui entraîne le changement de la topologie du réseau.

Par conséquent, les protocoles de routage conçus pour les RCSF doivent avoir une capacité d'auto-organisation qui les adapte à la distribution aléatoire des nœuds et à la topologie dynamique du réseau.

II.3.5 La mobilité

La plupart des architectures de communication des réseaux de capteurs supposent que les nœuds sont stationnaires. Cependant, la mobilité des nœuds capteurs ainsi que celle du nœud puits est parfois nécessaire dans plusieurs applications **[YE02]**.

Le routage des messages à partir, ou vers des nœuds mobiles est plus difficile puisque la stabilité des routes et l'imprévisibilité des positions des nœuds ajoute une difficulté importante, et nécessite une consommation d'énergie supplémentaire. Ceci s'explique, d'une part, par le fait que la mobilité des nœuds augmente le nombre de paquets perdus, et ainsi le nombre de retransmissions. D'autre part, afin de maintenir les routes à jour, il est nécessaire d'envoyer des messages de contrôle supplémentaires, induisant ainsi une consommation d'énergie additionnelle.

II.3.6 Modèles de transmission de données

Les paquets de données peuvent être transmis selon quatre modèles : continu, orienté-événement, orienté-requête et hybride **[EYA07]**.

Selon le modèle continu, les nœuds capteurs envoient continuellement les paquets de données suivant un taux de transmission prédéterminé. Dans ce cas, les protocoles de routage établissent au préalable les liens de communication.

Dans le modèle orienté-événement, les nœuds capteurs envoient les paquets de données lors d'une détection d'un événement (par exemple, détection d'incendie dans une forêt) et pour le modèle orienté-requête, ils les envoient lors d'une réception d'une requête générée par le nœud puits. Les applications orienté-événement et orienté-requête sont critiques et généralement intolérantes aux délais (c'est-à-dire temps réel). Suivant ces modèles, les protocoles de routage établissent les liens de communication à la demande.

Le modèle hybride concatène les trois modèles précédents.

II.3.7 Hétérogénéité

Généralement, les nœuds d'un RCSF sont homogènes ayant les mêmes capacités de calcul, de mémoire et de ressources énergétiques. Ces nœuds pourront être rapidement épuisés puisqu'ils réalisent plusieurs tâches à la fois comme le captage, le traitement et le routage de données. Pour y remédier, une solution envisagée par certaines applications consiste à intégrer des nœuds spéciaux plus puissants que les autres et qui seront chargés d'effectuer les tâches les plus coûteuses en termes de ressources énergétiques. Cependant, l'intégration d'un ensemble de nœuds hétérogènes dans un seul réseau impose de nouvelles contraintes liées au routage de données. En effet, les données récoltées par ces nœuds peuvent être soumises à des fortes qualités de service, et peuvent suivre des modèles de transmission de données différents. Par conséquent, la conception des protocoles de routage doit prendre en compte les différents types de nœuds, et les contraintes qui en résultent **[HNA06]**.

II.4 Métriques de routage

Cette section étudie les métriques communes utilisées pour mesurer l'efficacité des protocoles de routage. Un calcul de métrique est un algorithme qui traite un coût associé à un certain chemin de routage. Les protocoles de routage permettent aux nœuds de comparer les métriques calculées afin de déterminer les routes optimales à emprunter. Plus la métrique est optimale, plus le protocole de routage considère que la probabilité d'atteindre le nœud puits à travers ce nœud intermédiaire est grande **[IMT07]**. Plusieurs métriques peuvent affecter le routage en termes d'énergie, délai, longueur du chemin, etc. De plus, elles peuvent être considérées seules ou combinées (hybrides) **[PSY08]**.

II.4.1 Métriques pour la consommation énergétique

Les protocoles de routage utilisent cet ensemble de métriques pour minimiser la consommation d'énergie pendant le routage **[FAT07, LNA04]**. L'idée est de calculer l'énergie disponible (ED) pour chaque nœud du réseau et l'énergie nécessaire (EN) pour les transmissions des paquets entre une paire de nœuds.

Les routes entre les nœuds et le puits sont établies et chacune d'elles est caractérisée par la somme des ED des nœuds qui la constituent et par la somme des EN des liaisons qui la construisent. La consommation d'énergie suit plusieurs approches dont on peut citer :

II.4.1.1 Par considération de puissance

La route choisie est celle caractérisée par la somme des ED la plus élevée.

II.4.1.2 Par considération du coût

La route choisie est celle caractérisée par la plus petite somme des EN.

II.4.1.3 Par considération de puissance et du coût

Cette métrique est la combinaison des deux métriques précédentes. La route choisie est celle caractérisée par la plus petite somme des EN et la plus grande somme des ED.

II.4.2 Nombre de sauts

Les protocoles de routage utilisent cette métrique pour minimiser le nombre de sauts pendant le routage. L'idée est de calculer le nombre de nœuds intermédiaires pouvant être traversés lors d'une transmission d'un paquet du nœud source vers le nœud puits. La route choisie est celle qui contient un nombre minimum de nœuds (minimum de sauts) **[LNA04]**.

II.4.3 Perte de paquets

Les protocoles de routage utilisent cette métrique dans le but de minimiser le nombre de paquets de données perdus lors du transfert depuis une source vers une destination pendant le routage [**HNA06]**. L'idée est de calculer le ratio des paquets perdus et des paquets émis transitant dans le réseau. Autrement dit, on calcule le nombre de paquets perdus sur le nombre de paquets transmis lors d'une transmission. Dans le cas où le taux de perte de paquets est élevé, il est nécessaire de mettre en place des mécanismes qui permettent de minimiser les collisions.

II.4.4 Délai de bout-en-bout EED

L'EED (*End-to-End Delay*) est le temps moyen nécessaire pour qu'un paquet de données soit acheminé à partir de la source vers la destination [**HNA06]**. Cette technique est parmi les métriques les plus connues dans les réseaux sans fil. Les protocoles de routage l'utilisent pour minimiser le temps de propagation des paquets de données échangés pendant le routage.

II.5 Classification des protocoles de routage

Récemment, les protocoles de routage conçus pour les RCSF ont été largement étudiés. Les méthodes employées peuvent être classifiées suivant plusieurs critères qui sont illustrés dans le tableau suivant :

Type de classification	Classes
Paradigmes de communication	Centré-nœuds
	Centré-données
	Basé-localisation
	Basé-QoS
Topologie de réseaux	Plate
	Hiérarchique
Méthodes d'établissement de routes	Protocoles proactifs
	Protocoles réactifs
	Protocoles Hybrides

Tab. II-1 : Taxonomie des protocoles de routage. [ABD06, HNA06]

II.5.1 Classification selon les paradigmes de communication

Le paradigme de communication est déterminé par les contraintes sous lesquelles les nœuds du réseau sont interrogés. Dans les RCSF, il peut être classé comme étant centré-nœuds, centré-données, basés-localisation. Il existe également quelques protocoles basés sur la qualité de service, qui tentent de garantir certaines exigences des applications au moment du routage **[YAS06]**.

II.5.1.1 Centré-nœuds

Ce modèle est utilisé dans les réseaux conventionnels où il est important de connaître les nœuds communicants. Cependant, ce paradigme ne reflète pas la vision des RCSF quant à leurs applications où la donnée transmise est plus importante que l'émetteur. Néanmoins, le paradigme centré-nœuds n'est pas totalement écarté, car certaines applications nécessitent une interrogation individuelle des nœuds **[GSO02]**.

II.5.1.2 Centré-données

A la différence des réseaux ad hoc classiques, les nœuds d'un réseau de capteurs sont dépourvu d'une identification globale (tel que l'adressage IP). Cette absence d'identification explicite est due à certaines caractéristiques des réseaux de capteurs. Tout d'abord, l'obtention

et la gestion d'adresses nécessitent une étape de configuration qui peut être complexe, à cause du nombre très élevé de capteurs qui peuvent être déployés aléatoirement. De plus, étant donné la nature des applications des réseaux de capteurs, l'utilisateur ne s'intéresse pas à communiquer avec un nœud particulier du réseau mais envoie des requêtes, à tous les nœuds du réseau ou à une partie de celui-ci, afin d'identifier les nœuds concernés. Ces caractéristiques ont mené à la conception d'un nouveau type de protocoles appelé *centrés-données*. Dans ce dernier, le routage ne se fait pas en fonction d'une adresse de destination (comme c'est le cas pour les réseaux ad hoc), mais suivant les données disponibles au niveau des capteurs.

Les deux principales caractéristiques du routage centré-données sont :

II.5.1.2.a Adressage basé-attributs ou basé-localisation

Les requêtes envoyées par le nœud puits ainsi que les données de réponse correspondantes à celles-ci ne contiennent pas une adresse de destination, mais elles sont spécifiées par une description basée sur un attribut du phénomène capté ou sur sa localisation.

Par exemple, dans le cas d'un réseau chargé du contrôle de la température dans un immeuble, la requête qui demande le nombre de nœuds qui captent une température supérieure à 40°C utilise un adressage basé-attribut. Tandis que la requête qui veut connaître la température dans le sous-sol utilise un adressage basé-localisation. Un protocole de routage centré-données doit pouvoir identifier les capteurs concernés par les requêtes et acheminer les paquets vers le nœud puits.

II.5.1.2.b Agrégation des données

Dans un réseau de capteurs, plusieurs sources peuvent envoyer des données similaires. De plus, selon l'application, l'information captée par un seul nœud peut ne pas être très significative. Par exemple, pour un utilisateur désirant obtenir la température d'une région donnée, l'information fournie par un seul nœud n'est pas très représentative. La température globale de la région nécessite la collecte de toutes les données captées par les nœuds de cette région. Soulignons que le traitement de données est moins coûteux en consommation d'énergie que la communication. Il est donc plus avantageux de privilégier le traitement local des données en les agrégeant de sorte à réduire le nombre de transmissions et avoir des données plus significatives. Grâce à l'agrégation de données, une économie substantielle d'énergie ainsi qu'une amélioration de la qualité des données reçues par l'utilisateur peut être obtenue.

Les données captées sont agrégées si elles atteignent le même nœud au moment de leur acheminement vers le nœud puits. L'agrégation des données peut se faire en employant une fonction de combinaison des données telles que la suppression des redondances, le calcul du minimum, du maximum ou de la moyenne **[KRI02]**.

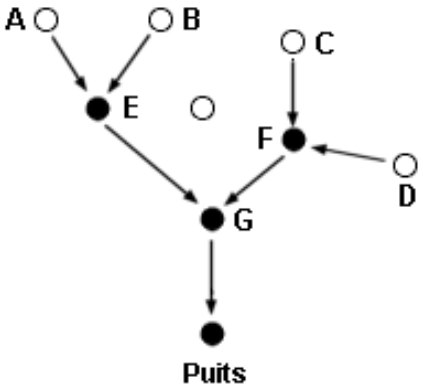


Figure II.1 : Agrégation des données

Dans l'exemple de la figure 2.1, le nœud E agrège les données provenant des nœuds A et B alors que le nœud F agrège les données provenant des nœuds C et D. Par la suite, G agrège les données de E et F.

L'agrégation de données peut également être réalisée d'une manière physique grâce à des techniques de traitement de signaux. Cette méthode est appelée "fusion de données". Elle consiste à employer certaines techniques telles que le *Beamforming* pour combiner les signaux reçus et réduire le bruit, afin de produire un signal plus précis **[AKK05]**.

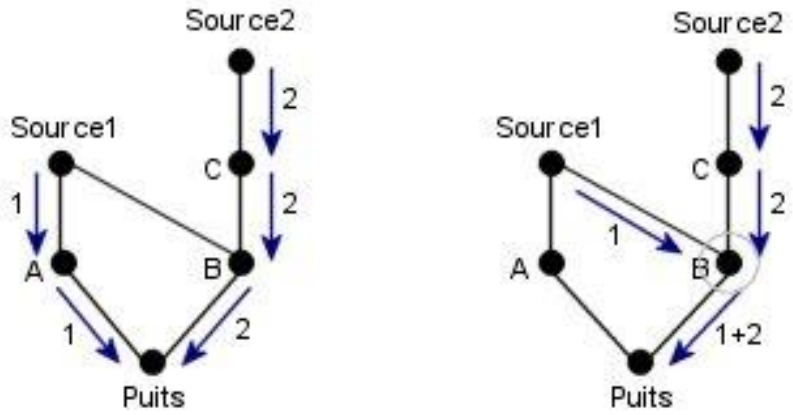


Figure II.2 : Routage centré-adresse et routage centré-donnée.

Une différence entre les protocoles centrés-adresse et les protocoles centrés-données réside dans la façon dont les données sont envoyées de la source vers le nœud puits. Dans un protocole centré-adresse, les données captées par chaque nœud sont envoyées au nœud puits d'une manière indépendante, à travers un chemin basé sur la route découverte par la requête. Dans les protocoles centrés-données, les nœuds choisissent la route en fonction du contenu du paquet de données, et peuvent privilégier les routes permettant de maximiser l'agrégation des données provenant des différentes sources **[KRI02]** (Figure II.2).

II.5.1.3 Basé-localisation

Ce paradigme est utilisé dans les applications où il est plus intéressant d'interroger le système en se basant sur la localisation des nœuds, et où on peut tirer profit des positions des nœuds pour prendre des décisions qui minimisent le nombre de messages transmis pendant le routage. Avant d'envoyer ses données à un nœud destination, le nœud source utilise un mécanisme pour déterminer sa localisation. Il est donc nécessaire de se pencher sur une solution de localisation géographique dont le degré de précision dépend de l'application visée. **[AHY08, AAJ07]**

Il existe deux techniques de localisation :

- absolue où on peut utiliser un système GPS (*Global Positioning System*) **[CHM07]**.
- relative où les nœuds sont localisés approximativement suivant la direction ou la durée lors d'une transmission **[IMO06]**.

II.5.1.4 Basé-QoS

Les protocoles de routage basés-QoS sont utilisés dans les applications qui ont des exigences temps-réel. Par exemple, dans le domaine de la sécurité, la détection d'intrusion doit être acheminée au plus bref délai vers le nœud puits. Ce type de protocoles essaye de répondre à quelques exigences de qualité de service (délais de transmission ou niveau de fiabilité) pendant le routage des données. Ceci, tout en maintenant un équilibre entre la qualité de service du routage et sa consommation énergétique. **[EYA07]**.

II.5.2 Classification selon la topologie du réseau

La topologie détermine l'organisation logique adaptée par les protocoles de routage afin d'exécuter les différentes opérations de découverte de routes et de transmission de données. Elle joue un rôle significatif dans le fonctionnement d'un protocole. La topologie peut être hiérarchique ou plate **[AJA04]**.

II.5.2.1 Topologie plate

Dans cette topologie tous les nœuds sont considérés homogènes et communiquent entre eux sans aucun autre intermédiaire, et seul le nœud puits est chargé de la collecte de données issues des différents nœuds capteurs afin de les transmettre vers les centres de traitement. Au cas où la destination (D) ne fait pas partie du voisinage de la source(S), les données seront transmises en utilisant des sauts multiples comme l'illustre la figure **II-3**.

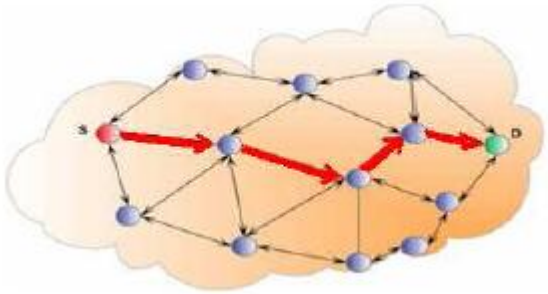


Figure. II-3: Topologie plate. [YAS07]

Les topologies plates sont caractérisées par la simplicité des algorithmes exécutés par les protocoles de routage. Et comme les RCSF souffrent des changements brusques de la topologie, une organisation plate permet la possibilité de construire différents chemins des sources vers le nœud puits.

Cependant, les réseaux plats présentent des inconvénients comme celui défini par le problème de *Hotspot*. En effet, tous les nœuds sont homogènes et il n'y a que le nœud puits qui est chargé de la récolte d'informations, et, ces dernières passent forcément par les nœuds qui entourent le nœud puits et qui seront de ce fait épuisés. Par ailleurs, les nœuds doivent accomplir plusieurs tâches en même temps ce qui pourrait rapidement épuiser leurs ressources énergétiques et dégrader ainsi les performances du réseau.
Plusieurs protocoles rentrent dans cette catégorie comme: *Directed Diffusion* **[LNA08]** et SAR (*Sequential Assignment Routing*) **[ABD07]**.

II.5.2.2 Topologie hiérarchique

Les réseaux hiérarchiques associent des rôles différents aux nœuds du réseau. Ils supposent des nœuds spéciaux plus puissants que les autres qui sont chargés d'effectuer les tâches les plus coûteuses en termes d'énergie afin d'alléger la charge sur les nœuds plus contraints en ressources énergétiques qui se consacrent uniquement au captage. De ce fait, des ensembles de ces derniers sont construits et gérés par les nœuds spéciaux appelés chefs d'ensembles. Dans ce cas, le routage devient plus simple, puisqu'il s'agit de passer par les chefs pour atteindre le nœud puits qui leur sont directement attachés.

Comme le montre la figure **II-3**, il existe deux configurations possibles pour les ensembles construits. Dans la première configuration, les membres d'un ensemble ne communiquent qu'avec leurs chefs de groupes, en obtenant ainsi un modèle basé sur les groupes. Dans la seconde, ils construisent des listes et les membres d'un ensemble utilisent d'autres nœuds comme passerelles appelés *Leaders* pour transmettre leurs données à leurs chefs en obtenant ainsi un modèle basé sur les listes (chaînes).

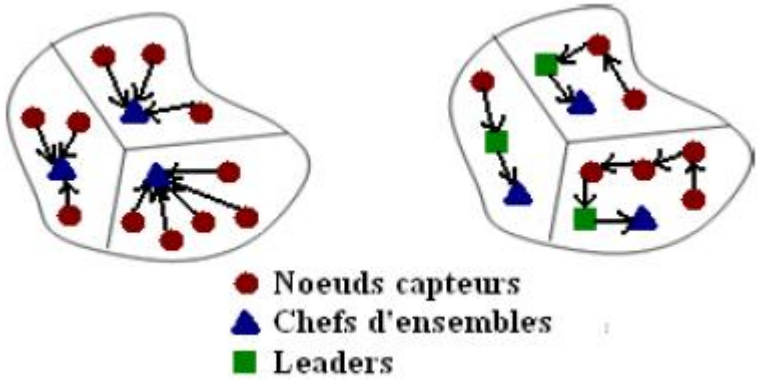


Figure. II-4: Configurations pour les RCSF découpés en ensembles. [ABD07]

L'avantage du routage hiérarchique est que les données d'un ensemble vont être combinées par le chef d'ensemble avant leur arrivée au nœud puits ce qui allège le travail de ce dernier, ainsi que celui des nœuds qui l'entourent. De plus, contrairement aux réseaux plats, un réseau hiérarchique possède une forte scalabilité. En effet, l'ajout des nœuds ne dégrade pas les performances du réseau car le réseau peut gérer seulement les nouveaux nœuds (par exemple, en les groupant et les associant à un chef d'ensemble) sans qu'il affecte tous les nœuds restants du réseau.

Cependant, les nœuds élus comme chefs d'ensembles consomment plus d'énergie que les autres nœuds. S'ils jouent toujours le rôle d'un chef d'ensemble, ils vont être épuisés à un moment donné. Par conséquent, le réseau va être divisé ce qui implique le découpage du réseau en secteurs inaccessibles. Parmi les protocoles qui utilisent cette topologie: LEACH (*Low-Energy Adaptive Clustering Hierarchy*) **[DJA08]**.

II.5.3 Classification selon la méthode d'établissement de routes

Suivant la manière de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage peuvent être séparés en catégories : protocoles proactifs, réactifs ou hybrides **[VIO08]**.

II.5.3.1 Protocoles proactifs

Les protocoles de routage proactifs établissent au préalable les meilleures routes pour chaque nœud vers toutes les destinations possibles. Ces protocoles maintiennent en permanence une vision globale de l'état du réseau grâce à une gestion périodique des tables de routage et l'échange des messages de contrôle. Ceci induit un contrôle excessif d'autant plus

qu'ils sont particulièrement utilisés pour les réseaux denses. De plus, ils présentent un autre inconvénient dû à la sauvegarde des routes même si elles ne sont pas utilisées.

II.5.3.2 Protocoles réactifs

Les protocoles de routage réactifs maintiennent des routes à la demande. Lorsque le réseau a besoin d'une route, une procédure de découverte 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 à rechercher les chemins, ce qui peut dégrader les performances des applications interactives.

II.5.3.3 Protocoles hybrides

Les protocoles hybrides combinant entre les deux techniques précédentes utilisent des méthodes proactives pour l'établissement de la route dans le proche voisinage (par exemple le voisinage à deux où trois sauts) et des méthodes réactives au delà de la zone de voisinage.

III.6 CONCLUSION

La propagation et l'acheminement de données dans un RCSF représentent une fonctionnalité très importante. Ils doivent prendre en considération toutes les caractéristiques du réseau afin d'assurer les meilleures performances du système. C'est pourquoi le routage dans les RCSF est un problème complexe. Ainsi, la conception d'un protocole de routage se fait selon des facteurs qui doivent être satisfaits pour atteindre une communication efficace.
Le taux de satisfaction de ces facteurs peut être mesuré par des métriques qui permettent de tester les performances du protocole de routage après sa réalisation.

Plusieurs protocoles de routage ont été proposés pour les RCSF grâce aux avantages qu'ils présentent. Bien que ces techniques de routage semblent prometteuses, ils restent encore de nombreux défis à relever. .Parmi ces protocoles, on distingue LEACH qui suit une architecture hiérarchique et qui est particulièrement intéressant pour les RCSF. En effet, il constitue un standard sur lequel est basée la conception de plusieurs protocoles de routage. LEACH constitue l'objet principal de notre étude. Ainsi, le chapitre suivant sera consacré pour son étude détaillée.

III.1 Introduction

Les protocoles de routage hiérarchiques sont considérés comme étant des protocoles très favorables en termes d'efficacité énergétique. Deux grandes approches sont dérivées de ce type de protocoles: l'approche basée sur les chaînes (*chaine-based approach*) dont l'idée de formation de chaînes a été proposée pour la première fois dans l'algorithme PEGASIS, et, l'approche basée sur les groupes (*cluster-based approach*). LEACH est considéré comme étant le premier protocole de routage hiérarchique basé sur la seconde approche. Il est aussi l'un des algorithmes de routage hiérarchiques les plus populaires pour les RCSF, proposés dans le cadre du projet μAMPS **[MOU05]**. Il combine l'efficacité en consommation d'énergie et la qualité de l'accès au média, et ce en se basant sur le découpage en groupes, en vu de permettre l'utilisation du concept de l'agrégation de données pour une meilleure performance en termes de durée de vie.

Dans la première partie de ce chapitre, nous expliquerons l'architecture, l'algorithme et les caractéristiques du protocole LEACH.

III.2 Protocoles MAC utilisés par LEACH

Pendant son fonctionnement, le protocole LEACH appelle certains schémas des protocoles MAC qui seront détaillés dans cette section pour mieux comprendre son déroulement. Les nœuds doivent avoir une certaine capacité de calcul pour supporter différents protocoles MAC. Comme les RCSF ont des caractéristiques distinctes de tout autre type de réseaux sans fil, les protocoles MAC conçus pour ces derniers ne sont pas toujours applicables dans les RCSF. Deux versions des protocoles MAC pour l'accès au media sont alors proposées pour les RCSF : l'accès aléatoire et l'allocation fixe **[LNA04]**.

III.2.1 Accès aléatoire

Les schémas à accès aléatoire sont à base de contention. Dans ces schémas, les nœuds qui possèdent des données à transmettre doivent essayer d'obtenir l'autorisation pour l'accès au media tout en réduisant les collisions avec les transmissions des données des autres nœuds.

Le schéma d'accès multiple avec surveillance de porteuse CSMA (*Carrier Sense Multiple Access*) sur lequel se base le protocole LEACH est l'un des schémas d'accès aléatoire **[ISA07b]**.

Lorsqu'un nœud veut transmettre un message, il examine le média pour vérifier s'il est libre ou occupé par un autre nœud. Dans le cas où le media est libre, ce nœud pourra émettre son message afin d'éviter les collisions. Cela dit, des nœuds peuvent émettre des données en même temps, ce qui mène à des collisions. Il est nécessaire donc que celles-ci soient détectées et que la récupération de données soit effectuée et que ces données soient retransmises.
Si les retransmissions se passent encore en même temps, d'autres collisions vont se produire.

Une solution à ce problème consiste à introduire que chaque nœud attende un délai aléatoire avant de retransmettre ses données, ce qui réduit la probabilité d'une autre collision. **[STE04]**

III.2.2 Allocation fixe

Les schémas à allocation fixe permettent d'allouer pour chaque nœud le media de transmission suivant des intervalles de temps (schéma TDMA) ou un schéma de codage particulier (schéma CDMA).
Étant donné que chaque nœud est attribué en exclusivité à un intervalle, il n'y a presque pas de collisions entre les données. Toutefois, les schémas à allocation fixe s'avèrent inefficaces lorsque tous les nœuds n'ont pas de données à transmettre. En effet, ces intervalles sont affectés à des nœuds qui n'ont pas besoin de les utiliser. **[PRE05]**

III.2.2.1 TDMA

Le schéma d'accès multiple à répartition de temps ou TDMA (*Time Division Multiple Access*) permet de diviser le temps en intervalles (*time-slot*) attribués à chaque nœud (voir figure III-1). Ainsi, un seul nœud a le droit d'accès au canal (il utilise toute la plage de la bande passante du canal), mais doit émettre ses données pendant les intervalles de temps qui lui sont accordés. **[STE04]**

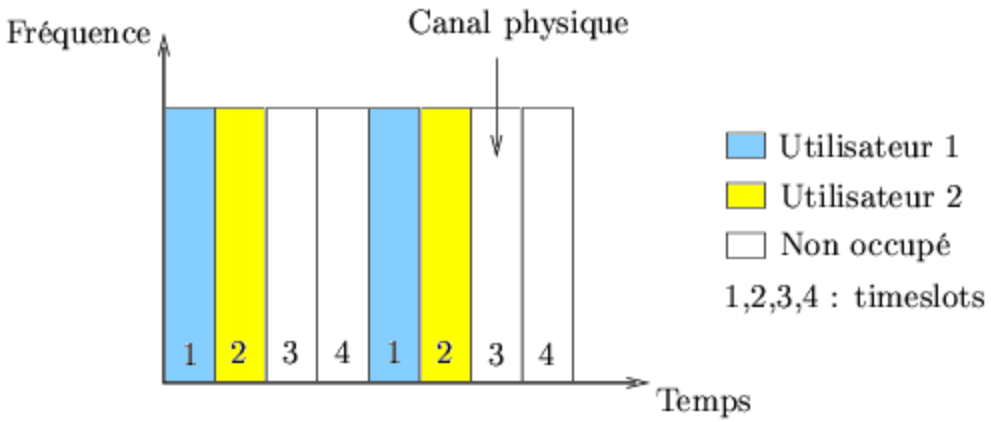


Figure. III-1 : Diagrammes représentant le protocole MAC TDMA.

III.2.2.2 CDMA

La méthode CDMA, ou accès multiple par division de codes, autorise l'allocation de la totalité de la bande de fréquences, de manière simultanée, à tous les utilisateurs.
Cette méthode utilise des techniques de codage adaptées au partage d'un même medium. Un code différent est affecté à chaque transmission. Cela fait qu'un signal utilisant un code interfère peu avec un autre signal utilisant un autre code.

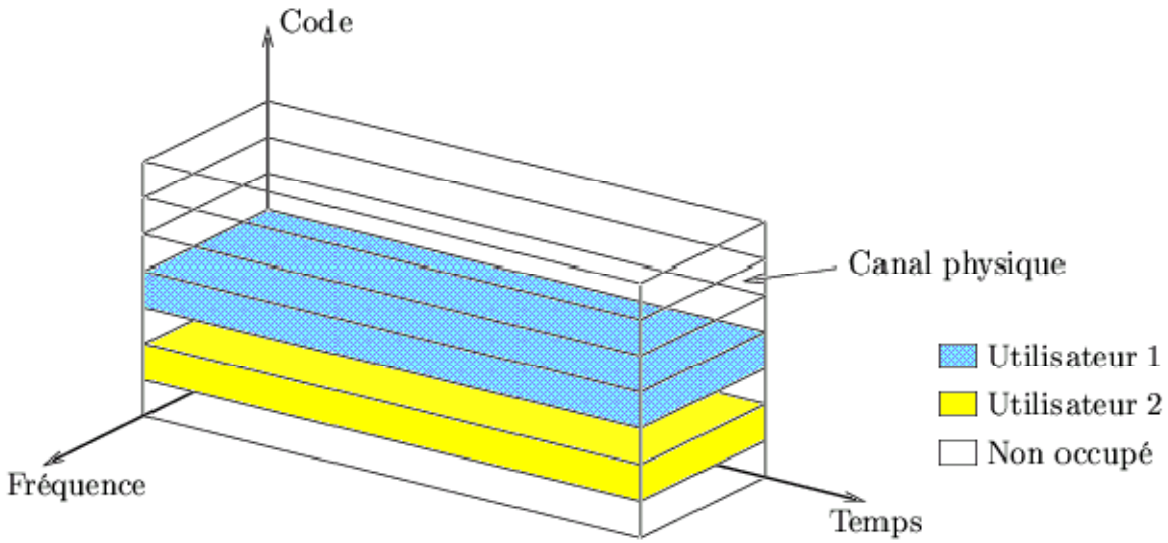


Figure. III-2 : Diagrammes représentant le protocole MAC CDMA.

III.3 Architecture de communication de LEACH

L'architecture de communication de LEACH consiste, de façon similaire aux réseaux cellulaires, à former des cellules basées sur l'amplitude du signal, et utiliser les têtes de cellules comme routeurs vers le nœud puits. Ces cellules sont appelées groupes (*clusters*), quant aux têtes : chefs de groupes (*cluster-heads* CH). Les chefs de groupes sont choisis de façon aléatoire selon un algorithme spécifique d'élection basé sur une fonction de probabilité qui prend en compte différents critères comme l'énergie disponible des nœuds.
Comme la figure IV-3 l'indique, les nœuds sont chargés de collecter des données, les envoyer à leurs CH qui les agrègent et transmettent, à leur tour, les résultats d'agrégation au nœud puits selon une communication unicast (à un seul saut).

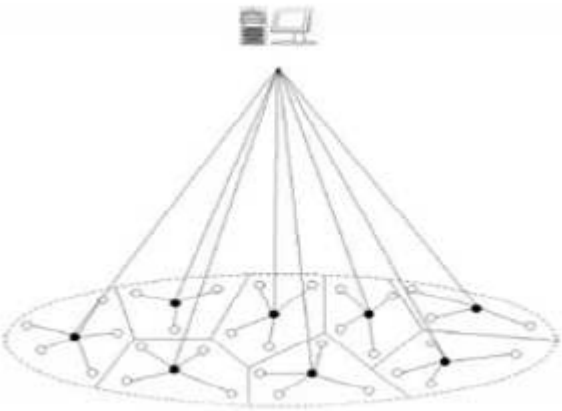


Figure. III-3 : Architecture de communication du protocole LEACH. [MOU05]

Les CH ont pour mission d'assurer les fonctions les plus coûteuses en énergie, à savoir la communication avec le nœud puits qui est supposé éloigné, ainsi que tous les traitements de données (agrégation, fusion et transmission de données) afin de réduire la quantité des données transmises. Ce dispositif permet d'économiser l'énergie puisque les transmissions sont uniquement assurées par les CH plutôt que par tous les nœuds du réseau. Par conséquent, LEACH réalise une réduction significative de la dissipation d'énergie **[SAM06]**.

III.4 Algorithme détaillé de LEACH

L'algorithme se déroule en « *rounds* » qui ont approximativement le même intervalle de temps déterminé au préalable. Chaque round est constitué d'une phase d'initialisation et d'une phase de transmission.

III.4.1 Phase d'initialisation

Comme l'indique la figure III-4, la phase d'initialisation est composée de trois sous-phases: d'annonce, d'organisation des groupes et enfin d'ordonnancement, et qui seront détaillée ci-dessous.

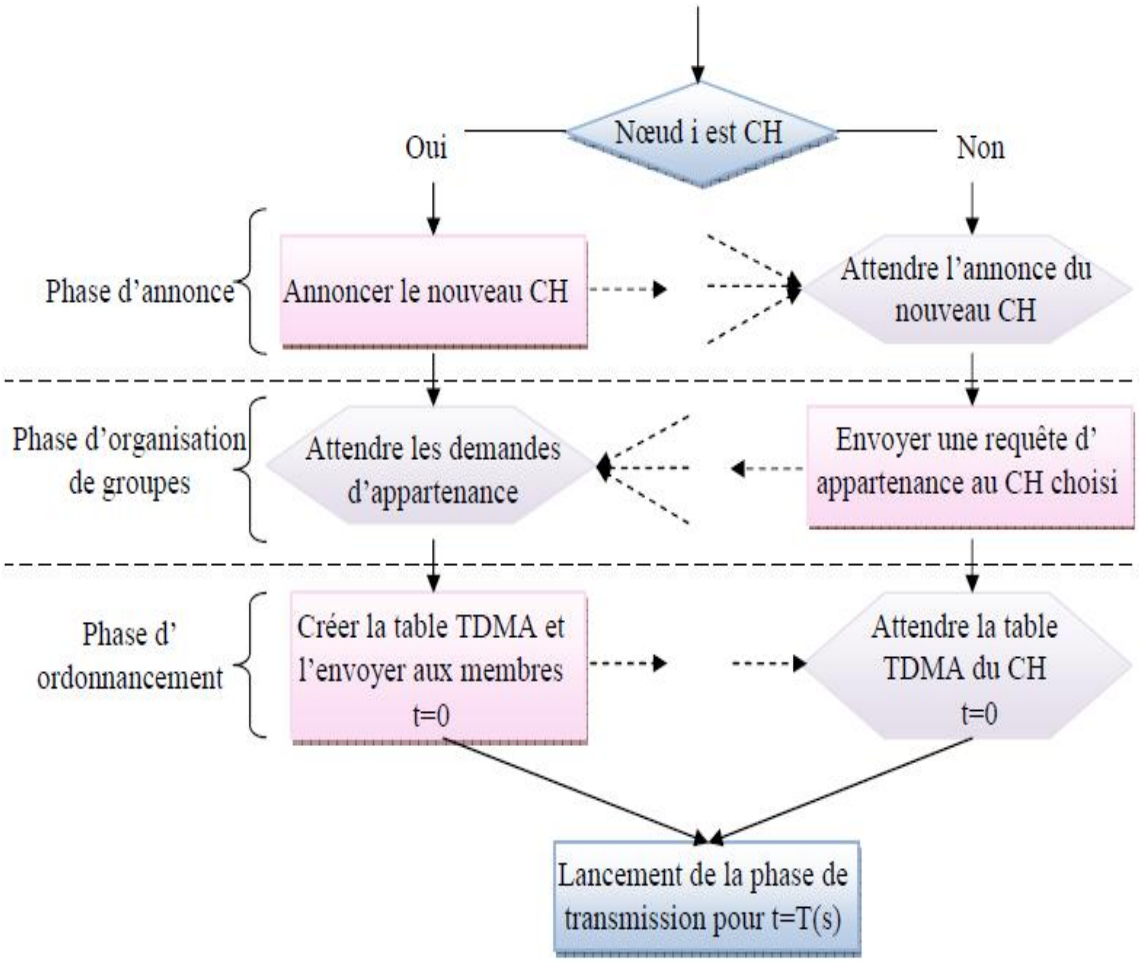


Figure. III-4 : Opérations de l'étape d'initialisation de LEACH [DJA08].

III.4.1.1 Phase d'annonce

Avant de lancer cette phase, on désire avoir un certain nombre de CH. Ce nombre, que l'on note K, est fixe et il est inchangé durant tous les rounds. On estime que le pourcentage optimal du nombre de CH désirés devrait être de 5% à 15% du nombre total de nœuds **[DJA08]**. Si ce pourcentage n'est pas respecté, cela mènera à une grande dissipation d'énergie dans le réseau. En effet, si le nombre de CH est très élevé, on aura un nombre important de nœuds (CH) qui se consacrent aux tâches très couteuses en ressources énergétiques. Ainsi, on aura une dissipation d'énergie considérable dans le réseau. De plus, si le nombre de CH est très petit, ces derniers vont gérer des groupes de grandes tailles. Ainsi, ces CH s'épuiseront rapidement à cause de travail important qui leur est demandés.

Cette phase commence par l'annonce du nouveau round par le nœud puits, et, par la prise de décision locale d'un nœud pour devenir CH avec une certaine probabilité Pi(t) au début du round r+1 qui commence à l'instant t. Chaque nœud i génère un nombre aléatoire entre 0 et 1. Si ce nombre est inférieur à Pi(t), le nœud deviendra CH durant le round r+1. Pi(t) est calculé en fonction de K et de round r **[WEN00]**:

$$\text{Nombre}(CH) = \sum_{i=1}^{N} Pi(t) = K$$

Où N est le nombre total de nœuds dans le réseau. Si on a N nœuds et K CH, alors, il faudra N/K rounds durant lesquels un nœud doit être élu seulement une seule fois autant que CH avant que le round soit réinitialisé à 0. Donc la probabilité de devenir CH pour chaque nœud i

est :
$$P_i(t) = \frac{\text{le nombre de CH désirés}}{\text{Le nombre de nœuds qui n'ont pas encore été élus CH durant les r rounds précédents}}$$

$$P_i(t) = \begin{cases} \dfrac{K}{N - k*(r \bmod N/k)} & : Ci(t)=1 \\ 1 & : Ci(t)=0 \end{cases} \qquad \dots (1)$$

Où Ci(t) égal à 0 si le nœud i a déjà été CH durant l'un des (r **mod** N/K) rounds précédents, et, il est égal à 1 dans le cas contraire. Donc, seuls les nœuds qui n'ont pas encore été CH, ont vraisemblablement une énergie résiduelle suffisante que les autres et ils pourront être choisis. Le terme $\sum_{i=1}^{N} Ci(t)$ représente le nombre total des nœuds éligibles d'être CH à l'instant t. Il est égal à :

$$\sum_{i=1}^{N} Ci(t) = N - K*(r \bmod N/K) \qquad \dots (2)$$

Utilisant l'équation (1) et (2), le nombre de CH par round est :

$$\text{Nombre}(CH) = \sum_{i=1}^{N} Pi(t) * Ci(t) = (K * (r \bmod N/K)) * \left(\frac{K}{N - k*(r \bmod N/k)}\right) = K$$

La probabilité Pi(t) est basée sur la supposition que tous les nœuds sont initialement homogènes et commencent avec la même quantité résiduelle d'énergie et meurent approximativement en même temps. Cependant, ceci pourrait être le cas juste après le déploiement, mais il n'est pas réellement valable après un certain temps. Alors, si l'énergie des nœuds diffère, il sera plus pratique que la probabilité Pi(t) soit en rapport avec l'énergie restante au niveau de chaque nœud. Cette probabilité sera donc égale à :

$$Pi(t) = \frac{Ei(t)}{Etotal(t)} K \qquad \dots (3)$$

Où Ei(t) est l'énergie résiduelle relative à chaque nœud i. Utilisant cette probabilité, le nœud avec une plus grande ressource d'énergie a une plus grande chance de devenir CH. Ainsi, le nombre de nœuds souhaités pour être CH dans chaque round est:

$$\text{Nombre(CH)} = \sum_{i=1}^{N} \text{Pi(t)} * \text{Ci(t)} = \left(\frac{\text{E1(t)}}{\text{Etotal(t)}} + \frac{\text{E2(t)}}{\text{Etotal(t)}} + \cdots \frac{\text{En(t)}}{\text{Etotal(t)}}\right) \text{K} = \text{K}$$

Les équations (2) et (3) seront égales si les nœuds commencent avec la même énergie. De plus, en utilisant l'équation (3), les nœuds requièrent des informations sur toute l'énergie disponible dans le réseau.

III.4.1.2. Phase d'organisation de groupes

Après qu'un nœud soit élu CH, il doit informer les autres nœuds non-CH de son nouveau rang dans le round courant. Pour cela, un message d'avertissement ADV contenant l'identificateur du CH est diffusé à tous les nœuds non-CH en utilisant le protocole MAC CSMA pour éviter les collisions entre les CH. La diffusion permet de s'assurer que tous les nœuds non-CH ont reçu le message. Par ailleurs, elle permet de garantir que les nœuds appartiennent au CH qui requière le minimum d'énergie pour la communication. La décision est basée donc sur l'amplitude du signal reçu; le CH ayant le signal le plus fort (i.e. le plus proche) sera choisi. En cas d'égalité des signaux, les nœuds non-CH choisissent aléatoirement leur CH **[WEN00]**.

Chaque membre informe son CH de sa décision. Une fois que le CH ait reçu la demande, il lui envoie un message d'acquittement Join- REQ.

III.4.1.3. Phase d'ordonnancement

Après la formation des groupes, chaque CH agit comme un centre de commande local pour coordonner les transmissions des données au sein de son groupe. Il crée un ordonnanceur (*schedule*) TDMA et assigne à chaque nœud membre un slot de temps durant lequel il peut transmettre ses données. L'ensemble des slots assignés aux nœuds d'un groupe est appelé frame. La durée de chaque frame diffère selon le nombre de membres du groupe.

Par ailleurs, afin de minimiser les interférences entre les transmissions dans des groupes adjacents, chaque CH choisit aléatoirement un code dans une liste de codes de propagation CDMA. Il le transmet par la suite à ses membres afin de l'utiliser pour leurs transmissions. **[SAC04]**

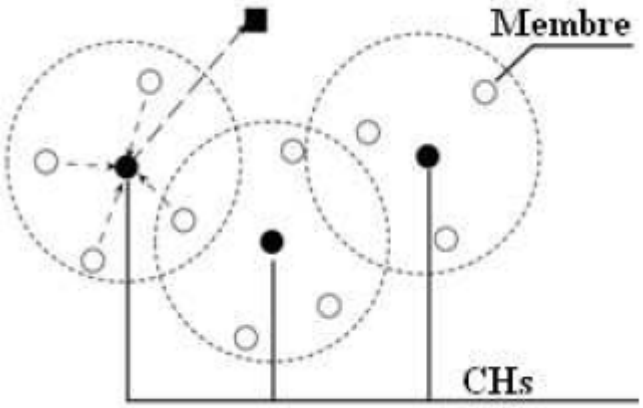


Figure. V-5 : Interférence lors d'une communication dans LEACH [ABD06].

III.4.2. Phase de transmission

Cette phase est plus longue que la phase précédente, et permet la collecte de données captées. En utilisant l'ordonnanceur TDMA, les membres émettent leurs données captées pendant leurs propres slots. Cela leur permet d'éteindre leurs interfaces de communication en dehors de leurs slots afin d'économiser leur énergie. Ces données sont ensuite agrégées par les CH qui les fusionnent et les compressent, et, envoient le résultat final au nœud puits.

Après un certain temps prédéterminé, le réseau va passer à un nouveau round. Ce processus est répété jusqu'à ce que tous les nœuds du réseau seront élus CH, une seule fois, tout au long des rounds précédents. Dans ce cas, le round est réinitialisé à 0.

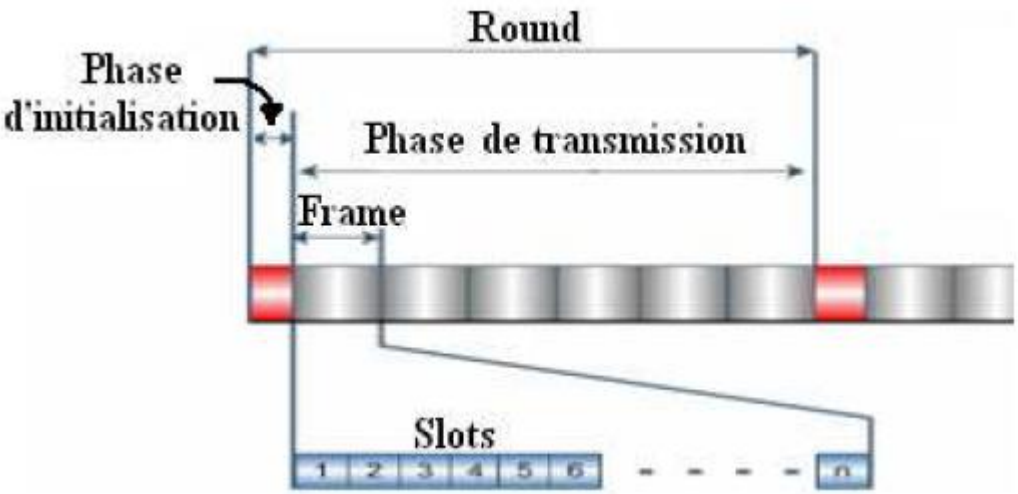


Figure. V-6 : Répartition du temps et différentes phases pour chaque round. [YAS07]

III. 5 Avantages et inconvénients de LEACH

Le protocole LEACH engendre beaucoup d'avantages en ce qu'il offre comme bonne manipulation de ressources du réseau en respectant plusieurs contraintes telle que la consommation d'énergie.

III.5.1 Avantages

- **Protocole auto-organisateur basé sur le groupement adaptatif**: LEACH est complètement distribué, autrement dit, les nœuds prennent leurs décisions de façon autonome et agissent de manière locale et n'ont pas besoin d'une information globale ni d'un système de localisation pour opérer de façon efficace. De plus, la collection de données est faite périodiquement (l'utilisateur n'a pas besoin de toutes les données immédiatement). Pour exploiter cette caractéristique, ce protocole introduit un groupement adaptatif, c'est-à-dire, il réorganise les groupes après un intervalle de temps aléatoire, en utilisant des contraintes énergétiques afin d'avoir une dissipation d'énergie uniforme à travers tout le réseau. **[SRA03]**

- **Rotation des rôles de chefs de groupes**: La rotation des rôles de chefs de groupes s'avère un facteur important pour l'organisation des nœuds. Ce rôle est épuisant en termes de d'énergie car les CH sont actifs tout au long de leur élection. Puisque le nœud puits est généralement loin du champ de surveillance, les CH diffusent une quantité plus importante d'énergie pour lui transmettre leurs données. Donc, si les CH sont choisis d'une manière fixe, leur énergie s'épuisera rapidement ce qui induit à leur défaillance. Par conséquent, tous les autres nœuds seront sans CH et donc inutiles. C'est pourquoi, les algorithmes de groupement (*clustering*) étudiés jusqu'ici adoptent la rotation du rôle de chefs de groupes. **[YAS07]**

- **Faible énergie pour l'accès au média**: Le mécanisme de groupes permet aux nœuds d'effectuer des communications sur des petites distances avec leurs CH afin d'optimiser l'utilisation du média de communication en la faisant gérer localement par un CH pour minimiser les interférences et les collisions.

III.5.2 Inconvénients

- On pourra ne pas avoir des CH durant un round si les nombres aléatoires générés par tous les nœuds du réseau sont supérieurs à la probabilité Pi(t).

- Les nœuds les plus éloignés du CH meurent rapidement par rapport aux plus proches.

- L'utilisation d'une communication à un seul saut au lieu d'une communication multi-sauts diminue l'énergie des nœuds.
- Le protocole LEACH ne peut pas être appliqué à des applications temps-réel du fait qu'il résulte en une longue latence.
- La rotation des CH permet de ne pas épuiser les batteries. Cependant, cette méthode n'est pas efficace pour de grandes structures de réseaux à cause de la surcharge d'annonces engendrées par le changement des CH, et qui réduit le gain d'énergie initial.
- Il n'est pas évident que les CH soient uniformément distribués. Donc, il est possible que les CH puissent être concentrés dans une partie du réseau. Par conséquent, certains nœuds n'auront pas des CH dans leurs voisinages.
- Le protocole LEACH n'est pas sécurisé. Aucun mécanisme de sécurité n'est intégré dans ce protocole. Ainsi, il est très vulnérable même aux simples attaques. Donc, un attaquant peut facilement monopoliser le réseau et induit à son disfonctionnement.
- Le protocole LEACH n'est pas conçu pour assurer l'agrégation des données comme un service privilégié, ce protocole utilise des fonctions simples pour l'agrégation.

III.6 Conclusion

Dans ce chapitre, nous avons présenté le protocole de routage hiérarchique LEACH qui suit une approche basée sur les groupes. Cette approche a montré son efficacité, comparée aux autres approches (par exemple, la topologie plate), en termes de consommation et de dissipation uniforme d'énergie prolongeant ainsi la durée de vie du réseau.

Nous avons vu que le protocole LEACH est soumis à certaines contraintes et suppositions qui engendrent toutefois des inconvénients. Par exemple, la communication unicast, établie entre le nœud puits et les CH et entre ces derniers et leurs membres, n'est pas toujours efficace par rapport à la communication multi-sauts.

IV.1 Introduction

Dans les réseaux sans fil, particulièrement dans les RCSFs, il existe plusieurs sources de congestion **[RCK10]** ; comme le débordement des mémoires tampon, les transmissions concurrentes, collision des paquets. Parmi les problèmes causés par la congestion, on peut citer : le retarder l'information, perte de paquets qui contiennent parfois les informations critiques, gaspillage de la bande passante. Il est facile de constater que la congestion dégrade les performances du réseau en l'empêchant de garantir certaines exigences de Qos comme le temps réel, le routage de bout en bout, la maximisation de la durée de vie du réseau. Ce problème motive le besoin de mise en place des mécanismes de contrôle de congestion pour améliorer la performance et prolonger la durée de vie du système **[RTHCC]**.

Notre objectif dans ce chapitre est de faire une étude bibliographique sur les protocoles ou techniques de contrôle de congestion proposées dans la littérature, afin de voir leur comportement dans certaines applications.

Pour cela, le reste de ce chapitre est organisé comme suit. Premièrement nous donnons quelques notions sur la détection et le contrôle de congestion dans les RCSFs. Ensuite nous présenterons quelques protocoles et techniques de contrôle de congestion avant la conclusion de ce chapitre.

IV.2. Définition et typologies de congestion

Dans un réseau informatique, la congestion est provoquée par les sources excédant le lien de communication ou la capacité (de stockage ou de traitement) des éléments du réseau. Pour cela, deux types de congestion pourraient se produire dans un RCSF **[RTHCC]**:

- la congestion au niveau du nœud (node-level congestion).
- la congestion au niveau du lien (link-level congestion).

La congestion au niveau du nœud, qui est répandue dans les réseaux conventionnels, est provoquée par le débordement des tampons dans le nœud et peut causer la perte et le retard des paquets.

Dans les RCSFs, le canal de communication sans fil est partagé par plusieurs nœuds en utilisant les protocoles de contrôle d'accès au média de la famille CSMA. Les collisions pourraient se produire quand les nœuds capteurs essaient de transmettre en même temps suite à la détection d'un un événement critique (incendie dans la forêt, séisme, etc.).

Dans cette situation, on parle de congestion au niveau du lien. Cette congestion conduit à un gaspillage de la bande passante et à un taux d'erreur élevé des paquets lors de leur réception.

La limitation de congestion suit généralement deux étapes **[RCK10]**: détection de congestion et contrôle de congestion. La détection exacte et efficace de congestion joue un rôle essentiel dans le contrôle de congestion dans les RCSFs **[RCK10]**.

Généralement il existe deux approches de contrôle de congestion **[RTHCC]**:

La gestion des ressources du réseau et la régulation du trafic.

La gestion des ressources du réseau essaie d'augmenter ses ressources (mémoire tampon, puissance d'émission de l'interface de communication, vitesse de traitement, etc.) pour atténuer la congestion quand elle se produit.

La régulation du trafic implique le contrôle de congestion par ajustement du taux de circulation des paquets afin qu'il s'adapte aux nœuds sources ou intermédiaires. La plupart des protocoles de contrôle de congestion existants appartiennent à ce type. La régulation du trafic peut être « bout en bout » ou « saut par saut ».Dans le bout en bout, la régulation s'effectue au niveau du nœud source pour simplifier la tâche aux nœuds intermédiaires ; il en résulte une réponse lente et dépend fortement du temps d'aller-retour. En « saut par saut » réponse est très rapide. Il est généralement difficile d'ajuster le taux de transmission des nœuds intermédiaires car ce taux dépend du protocole MAC et peut être variable.

IV.3 Classification des approches de contrôle de congestion

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

IV.3.1 Mécanisme de détection de congestion

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

IV.3.2 L'objectif du contrôle de congestion

Dans leur nature, les RCSFs sont orientés application (application specific).Donc, les protocoles de congestion seront différents selon l'application visée par un RCSFs où ils sont appliqués. Pour cette raison, les protocoles de contrôle de congestion sont aussi orientés application.

IV.3.3 Les mécanismes de contrôle de taux de transfert

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

IV.3.4 Equité et/ou QoS

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

Les différentes notions d'équité peuvent être utiles, selon l'application. Celles-ci peuvent être celles qui garantissent que tous les nœuds dans le réseau fournissent la même quantité de données (exemple, dans une simple application de collecte de données : simple data-gathering application), de l'équité maximum-minimum, de l'équité proportionnelle **[BHA05]**.

IV.3.6 D'autres métriques

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

IV.4 Quelques protocoles de contrôle de congestion dans les RCSFs

Dans le passé récent, de nouveaux protocoles de contrôle de congestion ont été proposés **[GBI00]**.Dans le paragraphe suivant nous allons faire une étude un peu détaillée sur certains d'entre eux.

III.4.1 Congestion detection and avoidance: CODA

CODA **[CSA03]** est une technique de contrôle de congestion pour les RCSFs qui comprend trois mécanismes :

1. *Détection de congestion basée sur le récepteur (Receiver-based congestion detection)*

L'occupation du tampon a été abondamment utilisée dans les algorithmes de détection de congestion traditionnels comme une mesure de niveau de congestion. Dans leur algorithme, les auteurs démontrent que l'occupation du tampon seule n'est pas une bonne mesure de congestion dans les réseaux sans fil à cause de la nature partagée du canal. La file d'attente peut se décongestionner potentiellement même si les paquets sont perdus en raison de la collision. Il est possible aussi pour les nœuds de déterminer la congestion en écoutant le canal et déterminer comment il est occupé/chargé .Cependant, cela peut avoir un coût énergétique significatif. Cependant, l'écoute continue encourt le haut prix d'énergie. Donc, la CODA utilise un plan d'échantillonnage qui active la surveillance du canal local surveillant seulement sous de certaines conditions, par exemple seulement quand le tampon d'envoi n'est pas vide, pour économiser l'énergie.

2. Open-loop hop-by-hop back-pressure

Quand le récepteur détecte une congestion, il envoie un message de suppression (une notification de congestion explicite), appelé « backpressure signal », en anglais, vers la source. Le message de suppression est envoyé à plusieurs reprises tant que l'état de congestion persiste. Les nœuds peuvent répondre à ce message en supprimant des paquets ou en réduisant leur taux. Le message de suppression peut se propager entièrement jusqu'à la source, ou atteindre seulement les nœuds intermédiaires selon leur état de congestion local.

3. Closed-loop multi-source regulation

Ce mécanisme de contrôle de congestion est utilisé par le puits pour intervenir dans la régulation des sources multiples, dans le cas où la congestion est persistante. Essentiellement, quand le taux de transmission d'une source excède le débit théorique maximum, S_{max}, la source informe le puits par un bit qu'elle met dans chaque paquet qu'elle transmet au puits tant que le taux de transmission reste supérieur à S_{max}.En réponse, le puits commence à envoyer les ACKs à la source jusqu'à ce qu'il détecte la congestion. Quand le puits détecte la congestion, il arrête d'envoyer les ACKs jusqu'à l'atténuation de la congestion, pour implicitement informer l'expéditeur de baisser son taux de transmission. En général, les sources maintiennent, diminuent, ou augmentent leurs taux selon la fréquence de réception des ACKs.

IV.4 .2 Fair rate allocation(FRA)

Fair rate allocation, ou allocation équitable de taux de transmission est une approche explicite au contrôle de congestion avec garantie d'équité proposée par Ee et Bajcsy **[CRB04]**. Le mécanisme de FRA comprend les trois démarches suivantes :

1. Déterminez le taux moyen, r, de transmission d'un paquet : En supposant que les paquets ont la même taille, le taux de transmission du paquet peut être estimé comme l'inverse de l'intervalle de temps de transmission d'un seul paquet. L'intervalle est mesuré à partir du moment où la couche transport envoie le paquet à la couche réseau jusqu'à au moment où la couche réseau signale que le paquet a été transmis.

2. Assignez le taux r aux nœuds en amont (c-à-d les nœuds fils dans l'arbre de collecte de données) : Le taux moyen de transmission de paquet est divisé par le nombre, n, de nœuds fils pour assigner le taux de génération de paquet de données comme $r\ data = r\ /n$

3. Pour calculer n, chaque nœud inclut la taille de son sous-arbre(le nombre de ses nœuds fils) dans un paquet et l'envoie au parent. Le parent décompte les nombre de ces descendants, y ajoute un (si le parent lui-même produit des données) et inclut le total dans le paquet avant de l'envoyer vers le puits.

Quand les files d'attente débordent ou sont au point de déborder, le nœud assigne un taux de génération de paquet inférieur aux nœuds qui sont en son amont.

Obtenir le taux du nœud parent r_{data_parent} par l'écoute du canal ou via un message de contrôle. Comparer r_{data} avec r_{data_parent} et propagé le plus petit taux aux nœuds du sous-arbre.

L'équité proportionnelle est obtenue en mesurant et en divisant le taux par le nombre de nœuds en aval. Il s'agit donc de l'équité proportionnelle Pour réaliser cela, chaque nœud maintient une file d'attente de type FIFO pour chaque nœud fils comme le montre la figure IV.1. Alors, un mécanisme de sélection probabiliste est employé pour mesurer le poids du choix des paquets. Le choix de la file d'attente à partir de laquelle le paquet sera transmis est proportionnel au nombre de nœuds entretenus par cette file d'attente.

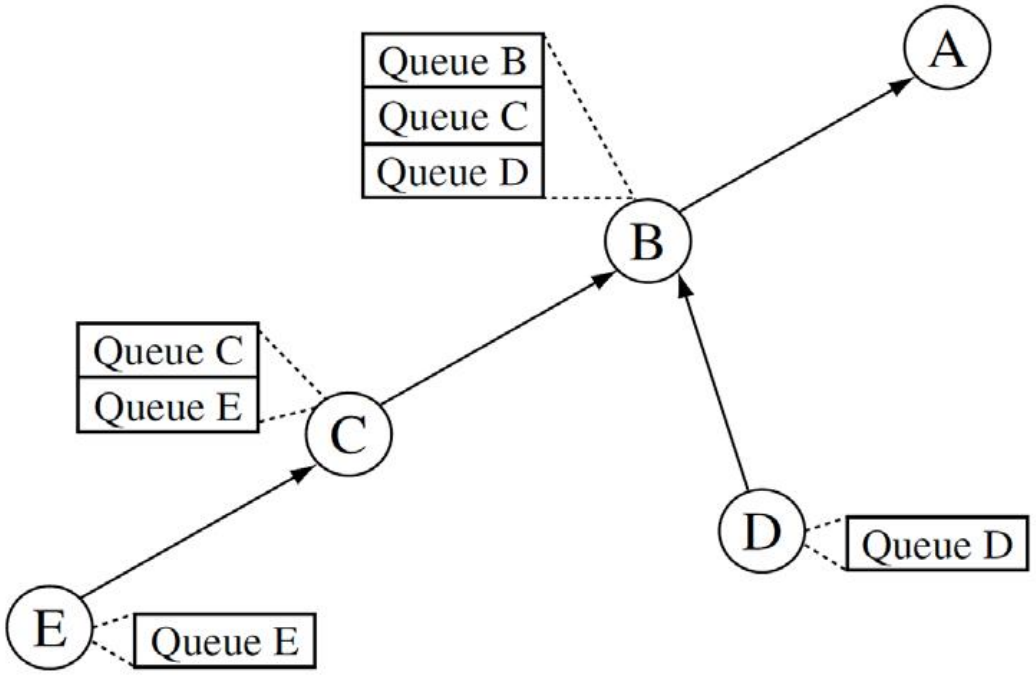


Figure IV.1 FIFO multiples pour assurer la délivrance équitable de données dans FRA.

IV.4 .3 Event-to-Sink Reliable Transport (ERST)

Le protocole ESRT **[CAP03]** se focalise sur l'ajustement du taux d'activité des nœuds sources afin d'assurer la fiabilité souhaitée par le puits, avec l'utilisation minimale de ressources. ESRT suppose que le puits est assez puissant pour atteindre tous les nœuds sources par diffusion. L'idée clé dans ESRT est que le puits ordonne les nœuds sources d'ajuster leur fréquence d'activité selon la fiabilité mesurée au niveau du puits et de l'état de congestion dans le réseau. ESRT piste deux paramètres : (1) l'indicateur d'intégrité, calculé par le puits; et (2) l'état actuel de congestion. Le puits calcule $\boldsymbol{\eta}$ pour la période i comme suit

$$\eta_i = \frac{r_i}{R_i}$$

Où r_i est la fiabilité d'événement observée et Ri est la fiabilité d'événement par le puits.

Pour informer le puits de l'état actuel de congestion, chaque nœud capteur contrôle la taille de sa file d'attente et met le bit de congestion dans le paquet à envoyé s'il constate que prochain paquet de données risque de causer un débordement de sa file d'attente.

En se basant sur ces paramètres, L'algorithme ESRT établit un diagramme de transition à cinq états comme le montre la figure **IV**.2 .Les états ont les significations suivantes :

- *No Congestion, Low Reliability (NC, LR):*

Le réseau n'est pas congestionné, mais la fiabilité observée est inférieure à la fiabilité souhaitée. Dans ce cas, les sources doivent augmenter leurs taux d'activité pour augmenter la fiabilité.

- *No Congestion, High Reliability (NC, HR):*

Le réseau n'est pas congestionné, mais la fiabilité observée est supérieure à la fiabilité souhaitée. Ainsi, le puits ordonne aux nœuds sources de réduire leurs taux d'activité prudemment, pour maintenir la fiabilité exigée, mais avec moins d'overheards.

- *Congestion, High Reliability (C,HR):*

Le réseau est congestionné et la fiabilité est supérieure à celle souhaitée. Dans ce cas, les nœuds doivent réduire leurs taux jusqu'à ce que la congestion soit résolue ou la fiabilité tombe en dessous du niveau souhaité.

- *Congestion, Low Reliability (C, LR):*

C'est le pire état possible, dans lequel ESRT réduit exponentiellement la fréquence d'activité pour alléger la congestion et potentiellement améliorer la fiabilité.

- *Optimal Operating Region (OOR):*

C'est la région d'exploitation optimale où le taux d'activité est suffisant juste pour atteindre la fiabilité souhaitée. Plus précisément,

$$1-\varepsilon \leq \eta_i \leq 1+\varepsilon$$

Où ε est une petite marge d'erreur utilisée pour assurer la stabilité. Le but d'ESRT est de maintenir toujours l'état du réseau dans OOR.

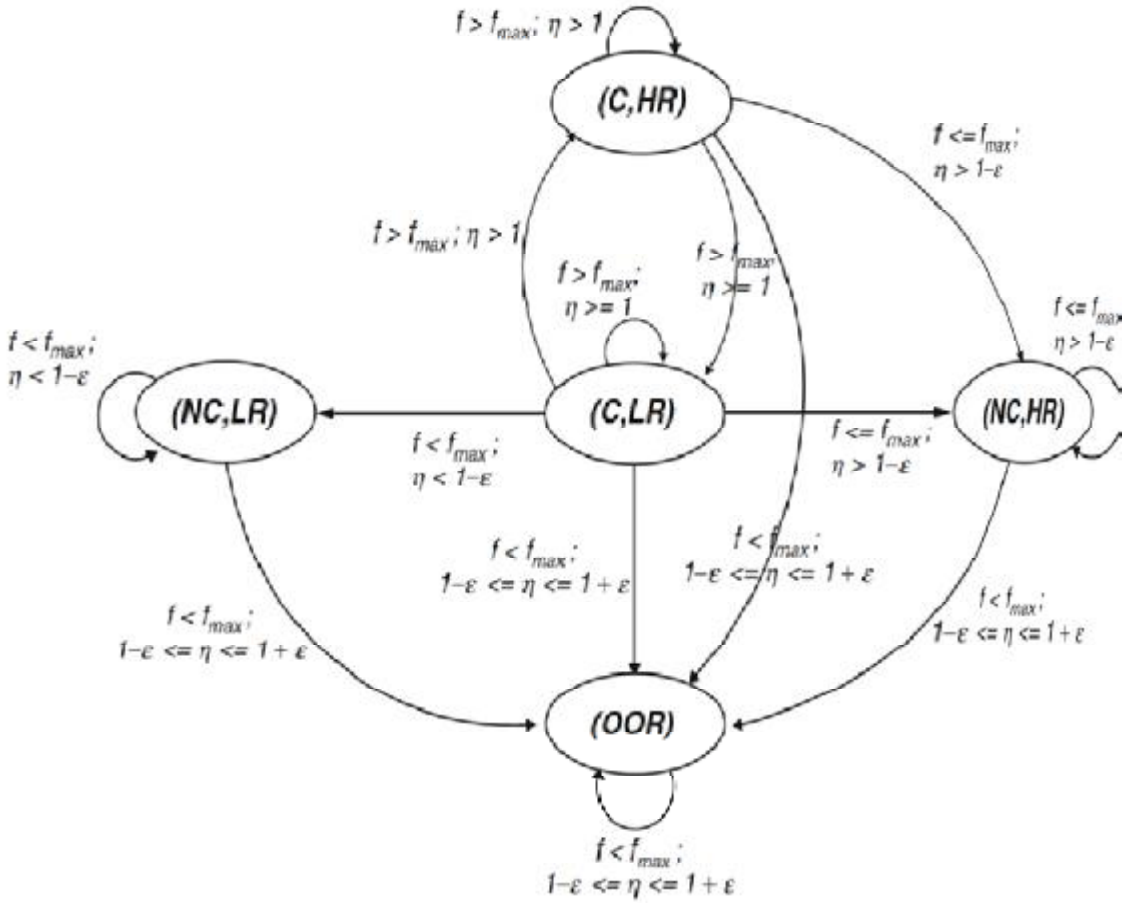


Figure **IV**. 2. Diagramme de transition du protocole ESRT.

IV.4.4 Autres protocoles

Le nombre élevé des protocoles de contrôle de congestion ne nous donne pas la possibilité de les explorer tous. Néanmoins nous avons pu détailler quelques uns dans les paragraphes précédents. Le choix des algorithmes détaillés dépend de leurs divergences et surtout de leur importance, selon notre point de vue, à résoudre ou à alléger le problème de congestion. Il existe d'autres algorithmes qui, la plupart, se basent sur ceux que nous avons a vus précédemment. Nous les résumons dans les paragraphes suivants.

Dans Fusion **[REJ04]**, basé sur CODA, les auteurs proposent un nouveau protocole MAC avec priorité (prioritezed MAC). L'idée principale de Fusion est qu'à la présence de congestion, le protocole MAC de la famille CSMAC classique qui alloue de façon équitable l'accès au canal de transmission à tous les nœuds n'est pas adapté.

Dans **[ZSJ02]**, les auteurs proposent le protocole IFRC (Interferance-Aware Fair Rate Control) qui prend en considération les interférences pour contrôler la congestion.

Dans **[AJO05]**, les auteurs partent de l'idée que dans un RCSF les événements captés peuvent être d'importances différentes pour proposer le protocole COMUT (Congestion Control Based on Importance of Data).

IV.5 Conclusion

Le partage du canal de transmission ainsi que la limite des ressources dont sont équipés les nœuds capteurs sont les principaux éléments qui peuvent conduire à un réseau de capteur congesté. La congestion, lorsqu'elle existe dans un réseau, dégrade les performances du système et peut causer beaucoup de problèmes.

Dans ce chapitre nous avons identifié quelques de ces problèmes parmi lesquels une perte des paquets, parfois contenant l'information critique (par exemple l'état de santé d'un patient), le gaspillage de la bande passante ainsi que d'énergie utilisés pour transmettre ces paquets perdus. Nous avons aussi étudié quelques les protocoles proposés pour détecter et/ou contrôler la congestion dans les RCSF.

Le point commun de ces protocoles est que, lorsqu'une congestion ou le premier signe de congestion apparait au niveau d'un nœud, ce dernier procède à une régulation du taux d'activité des nœuds sources des messages dont il est le destinataire.

Malgré que ces protocoles ont proposé une amélioration significative aux techniques classiques de contrôle de congestion qui se basaient presque uniquement sur le taux d'occupation du tampon de données au niveau du destinataire des messages pour détecter la congestion ,ces protocoles peuvent ne pas garantir la non perte des paquets dans le cas où une congestion au niveau d'un nœud persiste.

Dans le premier chapitre de ce mémoire, nous avons vu que l'une des propriétés d'un RCSF est qu'il est constitué par un grand nombre de nœuds déployés, parfois aléatoirement, dans une zone de captage et que selon la densité des nœuds du réseau et la nature du phénomène mesuré, ces nœuds sont susceptibles de capter et de transmettre la même donnée ou les données présentant un écart de valeur non significatif.

La transmission de la même donnée par les capteurs crée une redondance de données au niveau des nœuds intermédiaires utilisés pour transmettre ces données au nœud puits. Cette redondance, selon son intensité, peut être une source sérieuse de congestion. .

La technique utilisée pour réduire ou éviter cette redondance est connue sous l'appellation d'agrégation des données et va être décrite en détails dans le chapitre suivant.

V.1 Introduction

Selon leur mode de déploiement, les capteurs peuvent être disséminés dans une portion de leur zone d'intérêt où ils recueilleront les données fortement corrélées ou les mêmes données. La transmission de toutes ces données créerait une redondance de données au niveau des nœuds utilisés comme leurs points de relais vers le puits. Avec les moyens de traitement dont ils sont équipés, les capteurs peuvent réduire ces redondances, en combinant les paquets contenant les données redondantes. Cette technique s'appelle l'agrégation de données ou fusion de données.

Il ya deux raisons principales pour lesquelles un capteur ne devrait pas envoyer directement la donnée brute captée au nœud puits **[REJ04]**:La première raison est que le taux de transfert de données dans un RCSF est limitée. La deuxième est que la communication est plus consommatrice en énergie que le traitement. En effet, il a été montré dans plusieurs publications scientifiques que la transmission d'un seul bit est équivalente, en termes d'énergie, à l'exécution d'environ 1000 instructions **[ACC07]** et que cette valeur augmente avec la portée du module radio. Il convient donc de réduire cette énergie en agrégeant les données dans leur routage.

Le problème des données redondantes est que non seulement elles sont la source de gaspillage de l'énergie mais aussi dans la plupart des cas elles sont source de congestion dans le réseau. Un bon algorithme d'agrégation des données pourrait donc réduire ou éviter considérablement cette congestion.

L'objectif de ce chapitre est de passer en revue de certains algorithmes d'agrégation de données proposés dans le passé est de voir s'ils peuvent être utilisés pour réduire la congestion dans quelques genres d'applications.

Le reste de ce chapitre est organisé comme suit :

En premier lieu nous verrons quelques notions générales sur les techniques d'agrégation de données. Ensuite nous ferrons une étude sur les techniques d'agrégation existant à travers une certaine classification et nous finirons par une conclusion.

V.2 Définition

L'agrégation données est un processus global de collecte et de routage d'information par un réseau multi-sauts, en traitant des données aux nœuds intermédiaires avec l'objectif de réduire la consommation de ressources et ainsi augmenter la durée de vie du réseau **[EMGM]**.

V.3 Performances et limites d'une technique d'agrégation

Plusieurs études théoriques fournissent des limites sur la performance de techniques d'agrégation de données dans le réseau et aident ainsi la conception d'algorithmes convenables **[EMGM]**. L'efficacité de ces algorithmes dépend de la corrélation entre les données produites par de différentes sources d'information (les unités de captage). Une telle corrélation peut être spatiale, quand les valeurs produites par les capteurs voisins sont rattachées, temporelles, quand les lectures de capteurs changent lentement au fil du temps, ou sémantique, quand les contenus de différents paquets de données peuvent être classifiés sous le même groupe sémantique (par ex, les données produites par les capteurs placés dans la même pièce). Les augmentations d'agrégation de données dans le réseau peuvent être le mieux démontrées dans le cas extrême quand les données produites par de différentes sources peuvent être combinées dans un paquet simple (par ex, quand les sources produisent des données identiques.

V.4 Approches d'agrégation de données

Dans un RCSF on distingue deux approches d'agrégation de données **[EMGM, KKN11**] :agrégation sans et avec réduction de la taille des données .

- *Agrégation avec réduction de la taille* ***(****In-network aggregation with size reduction)* ***:*** fait allusion au processus de combinaison des données venant de différentes sources pour réduire la quantité de données à envoyer dans le réseau. Comme exemple, supposez qu'un nœud reçoit deux paquets de deux différentes sources contenant les températures localement mesurées. Au lieu d'envoyer les deux paquets, le capteur peut calculer la moyenne des deux lectures et l'envoyer dans un seul paquet.

- *Agrégation sans réduction de la taille (In-network aggregation without size reduction):* fait allusion au processus de fusion des paquets venant de différentes sources dans le même paquet sans aucun traitement sur les données. Supposez par exemple qu'un nœud reçoit deux paquets contenant les données qui décrivent les phénomènes physiques différents, par exemple, l'humidité et l'acidité du sol. Ces deux valeurs ne peuvent pas être traitées ensemble mais elles peuvent être transmises dans un seul paquet simple et réduire ainsi l'overhead.

L'avantage de la première approche est qu'elle permet de réduire au maximum la quantité de données échangées. Son principal inconvénient est que après l'opération d'agrégation, il n'est pas toujours possible de reconstituer tous les paquets originaux.La seconde approche, par contre, préserve les paquets originaux (c.à.d. au niveau du puits, les paquets originaux peuvent être reconstruits). Le choix de l'approche à utiliser dépend de

plusieurs facteurs dont : type d'application, taux de transfert de données, les caractéristiques du réseau.

V.5 Les éléments de base de l'agrégation de données

Les techniques d'agrégation dans le réseau exigent trois éléments fondamentaux **[EMGM]** : un protocole de routage convenable, les fonctions d'agrégation efficaces et une façon efficace de représenter les données, voir figure IV.1 ci-dessous. Dans le reste de cette section nous décrivons chacun de ces aspects.

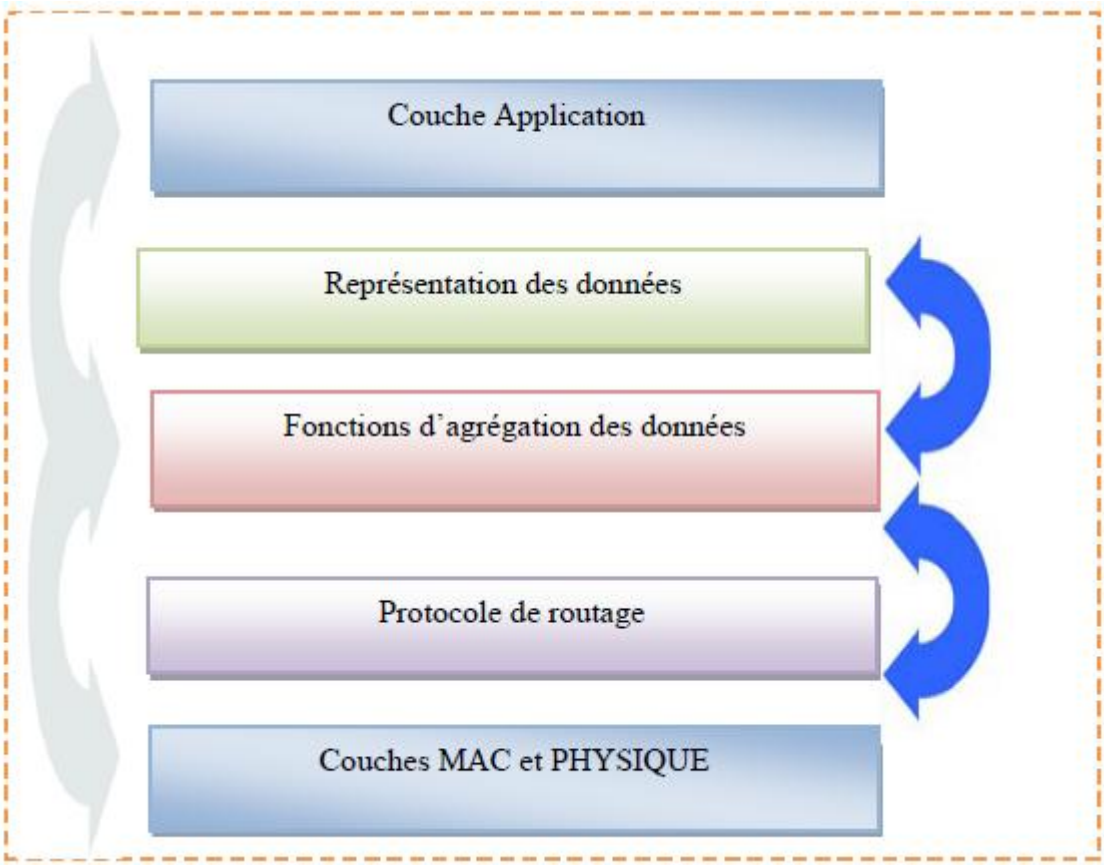


Figure V.1 Relation entre la technique d'agrégation et d'autres couches de la pile

Notons que dans la plupart des cas, la technique d'agrégation interagit avec les couches application, Physique et MAC, comme le montre la figure ci-dessus.

V.5.1 Fonctions d'agrégation de données.

L'une des fonctionnalités les plus importantes que les techniques d'agrégation dans le réseau devraient fournir est la capacité de combiner des données venant de différents nœuds.

Il y a plusieurs types de fonctions d'agrégation et la plupart d'entre elles sont fortement dépendantes de l'application cible. Néanmoins, nous pouvons identifier quelques paradigmes communs pour leur classification **[EMGM, KKN11]** :

- *Insensibles à la duplication Vs sensible à la duplication*

Dans un réseau de capteur sans fil, un nœud intermédiaire peut recevoir des copies multiples d'une même information. Dans ce cas, il arrive que cette même information soit considérée plusieurs fois lors de l'agrégation, la fonction d'agrégation utilisée est sensible à la duplication (duplicate sensitive), le résultat final dépendra du nombre de fois que la même valeur a été considérée. Dans le cas contraire, la fonction d'agrégation est dite insensible à la duplication (duplicate insensitive).Par exemple, les fonctions telles que MIN et MAX qui calculent respectivement la valeur minimale et maximale des données reçues au niveau d'un nœud sont insensibles à la duplication tandis que les fonctions telles que SUM et AVG, qui calculent respectivement la somme et la moyenne de ces mêmes données sont sensibles à la duplication.

- *Sans perte Vs Avec perte*

L'agrégation de données peut se faire avec ou sans perte d'information. L'agrégation avec perte ne permet pas de reconstruction parfaite mais l'agrégation sans pertes garantit une récupération complète de toutes les données de capteur individuelles à la station de base (puits) **[TJS05]**.

De bonnes fonctions d'agrégation pour les réseaux de capteur sans fil ont besoin de satisfaire des besoins supplémentaires. En particulier, elles devraient tenir compte du traitement très limité et des capacités d'énergie des nœuds intermédiaires et devraient être donc implémentées au moyen des opérations élémentaires.

V.5.2 Protocoles de routage avec agrégation des données

L'ingrédient le plus important pour l'agrégation dans le réseau est un protocole de routage bien conçu **[EMGM]**. Comparativement au routage classique, la prise en compte d'agrégation des données exige de nouveaux paradigmes pour acheminer les données .Par exemple, les techniques d'agrégation dans le réseau peuvent exiger une forme de synchronisation parmi les nœuds. En particulier, la meilleure stratégie d'un nœud donné n'est pas toujours d'envoyer des données aussi tôt que c'est disponible. L'attente d'information venant des nœuds voisins peut mener à de meilleures techniques d'agrégation des données et, ainsi, améliorer la performance. Selon la stratégie utilisée pour choisir le moment où un nœud doit retransmettre la donnée agrégée, nous pouvons distinguer trois types d'agrégation : [70] : Agrégation périodique simple, Agrégation périodique par saut et Agrégation périodique par saut ajusté.

- *Agrégation périodique simple (Periodic simple aggregation) :*

Chaque nœud doit attendre une période de temps prédéterminé, pour agréger tous les paquets reçus avant de transmettre le résultat de l'agrégation.

- *Agrégation périodique par saut (Periodic per-hop aggregation) :*

Est similaire à l'agrégation périodique simple, la seule différence est que la donnée agrégée est transmise aussitôt que le nœud entend de tous ses enfants(les nœuds fils dans l'arbre de collecte de données) .Cela nécessite que chaque nœud connaisse le nombre de ces enfants. En plus, un temps mort (timeout) est utilisé dans le cas où il y a une perte de paquets de quelques nœuds enfants.

- *Agrégation périodique par saut ajustée (Periodic per-hop adjusted aggregation)*

Ajuste le temps mort d'un nœud, après lequel il envoie les données agrégées, selon la position du nœud dans l'arbre se de collecte de données.

Notons que le choix de la stratégie à utiliser affecte fortement la conception d'un protocole de routage **[IKO04,XZK06]**.

Même si nous avons présenté les protocoles de routages et les techniques d'agrégation dans deux chapitres différents, notons que ces deux notions sont fortement liées. Dans le deuxième chapitre, nous avons vu les protocoles de routage dans le cas général. Un certain nombre d'entre eux réalisent l'agrégation des données. Nous pouvons citer comme exemples

Les protocoles: LEACH, Directed Diffusion, .Etant données que ces protocoles ne sont pas conçus pour assurer l'agrégation des données comme un service privilégié, ces protocoles utilisent des fonctions simples. Dans ce qui suit, nous essayons de donner un aperçu sur quelques protocoles de routages avec agrégation des données en les classifiant quatre catégories : hiérarchique, basés sur les clusters, multi-chemins et hybrides.

V.5.2.1 Approche hiérarchique

La plupart des travaux faits sur l'agrégation de données dans les RCSFs ont proposé des solutions exploitant une structure hiérarchique (arborescente) **[KMY05]**.En effet, la façon la plus simple d'agréger des données venant des sources vers le puits est d'élire quelques nœuds spéciaux qui travaillent comme les points d'agrégation et définissent une direction préférée à être suivie en envoyant des données.

. Dans les paragraphes suivants, nous considérons les principaux algorithmes de routage basées sur les arbres d'agrégation de données.

1. Tiny AGregation (TAG)

TAG **[SMJ02]** est un protocole data centric basé sur les arbres d'agrégation et est spécifiquement conçu pour l'application de surveillance .Cela signifie chaque nœud devrait produire l'information périodiquement. Donc, il est possible de classifier TAG comme un protocole à Agrégation périodique par saut ajustée.TAG se compose de deux phases : une phase de *distribution* (distribution phase), dans laquelle les requêtes sont disséminées et une phase de *collecte* (collection phase), où les valeurs agrégées sont continuellement routées vers le haut.

Pendant la phase de distribution, le puits diffuse un message en demandant aux nœuds de s'organiser dans un arbre de routage et envoie ensuite ses requêtes. Dans chaque message il y a un champ spécifiant le niveau (level), ou la distance depuis racine, qui est incrémenté à chaque fois qu'un nœud reçoit un message et le rediffuse aux autres nœuds voisins. Le niveau du puits est égal à zéro. Ce processus continue jusqu'à ce que tous les nœuds aient été assignés identificateur et un parent.

TAG adopte la sélection et l'agrégation offertes par les langages d'interrogation de bases de données (SQL). Par conséquent, les requêtes de TAG ont la forme suivante :

SELECT {agg(expr), attrs} from SENSOR
WHERE {selPreds}
GROUP BY {attrs}
HAVING {havingPreds}
EPOCH DURATION i

En pratique, le puits envoie une requête, où il spécifie les quantités qu'il veut recueillir (le champ *attrs*), comment celles-ci doivent être agrégées (*agg (expr))* et les capteurs qui devraient être impliqués dans l'extraction de données. Cette dernière demande est spécifiée par les clauses WHRERE, GROUP BY et HAVING [76]. Finalement, un champ de durée *EPOCH* spécifie le temps (en secondes) que chaque capteur devrait attendre avant d'envoyer de nouvelles lectures de captage. Cela signifie que les utilisées pour calculer le résultat de l'agrégation appartiennent toutes au même intervalle de temps, ou epoch.

Pendant la phase de collecte de données, en raison de la structure arborescente, chaque parent doit attendre des données de tous ses enfants, pendant un temps égal à epoch, avant qu'il puisse envoyer le résultat de l'agrégation à son parent. Les *epochs* sont divisés en intervalles plus courts appelés slots de communication. Le nombre de ces slots est égal à la profondeur maximum de l'arbre de routage et l'agrégation de données est exécutée par tous les nœuds intermédiaires.

Exemple de requête utilisée dans TAG :

```
SELECT AVG(temperature),chambre FROM capteurs
WHERE etage = 10
GROUP BY chambre
HAVING AVG(temperature) > 25
EPOCH DURATION 60s
```

Cette requête partitionne les capteurs se trouvant au dixième étage d'un immeuble suivant les chambres dans lesquelles ils se trouvent. La requête renvoie toutes les chambres dans lesquelles la température moyenne est supérieure à 25 unités de température. Les résultats mis à jour sont envoyés après chaque minute (60secondes).

2. Directed Diffusion

Dans Directed Diffusion, l'agrégation de données est exécutée, quand les données sont envoyées au puits, au moyen des méthodes convenables, qui peuvent être choisies selon les exigences de l'application. L'arbre de collecte de données (renforcement des chemins) doit être périodiquement rafraîchi par le puits. Cela peut être cher en cas de topologies dynamiques.

V.5.2.2 Approche basée sur les clusters

Dans les protocoles bases sur les clusters tels que LEACH, l'agrégation est exécutée par les cluster-heads qui communiquent directement avec la station de base. Pour distribuer la consommation de l'énergie de manière équitable sur tous les nœuds.

V.5.2.3 Approche multi-chemins

L'idée principale de cette approche est que chaque nœud peut envoyer à ses (si possible) multiples voisins en exploitant la nature broadcast du média sans fil. Une structure qui convient bien à cette approche s'appelle topologie en anneaux où les nœuds capteurs sont divisés en plusieurs niveaux selon le nombre de sauts les séparant du puits. L'agrégation de données est exécutée sur les chemins multiples puisque les paquets se déplacent niveau par niveau vers le puits (voir figure V.2).

Dans ce qui suit, nous allons voir Synopsis Diffusion qui appartient à cette classe de protocoles.

Dans **Synopsis Diffusion [SPZ04]** la topologie de diffusion de données est organisée en anneaux concentriques autour du puits. Synopsis Diffusion comprend deux phases : la phase de distribution des requêtes (distribution of the queries) et la phase d'extraction de données (data retrieval). La topologie en anneau est formée quand un nœud envoie une requête dans le réseau. En particulier, deux différentes structures, énumérées ci-dessous, peuvent être prises

en compte. Le premier type de topologie consiste en structure d'anneau simple. Pendant la phase de distribution des requêtes, les nœuds du réseau forment un ensemble d'anneaux autour du nœud émetteur de la requête *q*, qui est le seul capteur appartenant à l'anneau R_0. Un nœud est dans l'anneau R_i s'il est à i sauts du nœud q. Le deuxième type de topologie a quelques améliorations qui le rendent plus robuste que le précédent et capable de faire face aux changements dans le réseau. Cette topologie s'appelle *anneaux adaptatifs* (adaptative rings). La phase de distribution ne change pas mais cette fois ci un nœud *n* dans l'anneau *i* garde la trace du nombre de fois, n_{ov}, que les transmissions de n'importe quel nœud n_{i-1} dans l'anneau *i-1* ont inclus ses propres données pendant quelques derniers epochs. Si le novembre est petit, *n* essaie de trouver un meilleur anneau pour avoir plus de ses propres données incluses dans les transmissions ultérieures.

Dans l'exemple de la figure IV.2, les données générées au nœud A peuvent atteindre le puits par sept chemins :{A, B, F, I, S}, {A, B, F, H, S}, {A, B, F, G, H, S}, {A, C, D, E, I, S}, {A, C, F, H, S}, {A, C, F, I, S} et {A, C, G, H, S}.

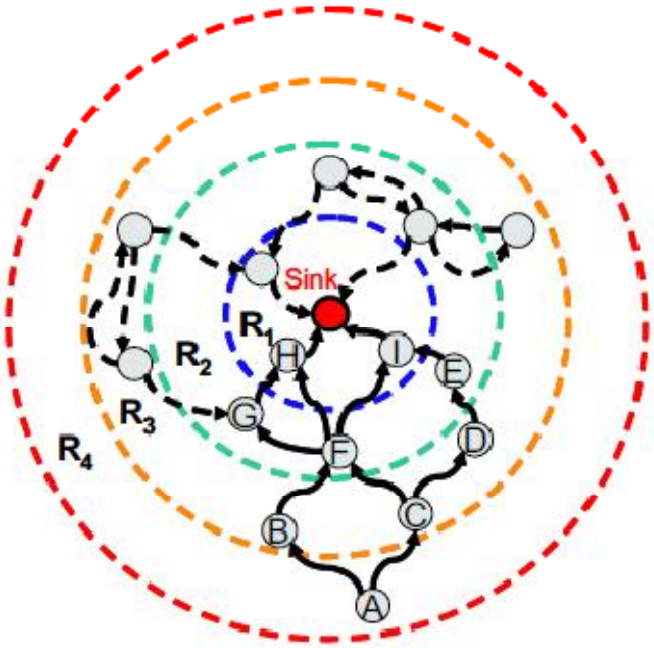


Figure V. 2 Exemple d'une technique d'agrégation utilisant une structure en anneau

Notons que comme la caractéristique principale de Synopsis Diffusion est que les données peuvent être transmises via les multiples chemins, un nœud peut recevoir des copies de la même information. Cela peut affecter le résultat d'agrégation, surtout quand les fonctions d'agrégation sont sensibles à la duplication. Ce problème est adressé par les auteurs dans **[SPZ04]** en proposant des fonctions et des structures de données convenables, qu'on verra dans la suite de ce chapitre.

V.5.2.4 Approche hybride

Pour profiter des avantages de deux approches (hiérarchique et multi-chemins), il est possible de définir des approches hybrides. L'exemple typique est présenté dans [78] et est expliqué ci-après.

Dans le protocole **Tributaries and Deltas [ASP05]** les structures d'agrégation des données peuvent circuler simultanément dans de différentes régions du réseau. L'idée consiste en ce que sous les taux faibles de perte de paquets, un arbre d'agrégation de données est la structure la plus convenable. D'autre part, quand ces taux deviennent, une approche multi-chemins peut être la meilleure. Pour cela, les nœuds sont divisés en deux catégories : les nœuds utilisant l'approche hiérarchique pour envoyer les paquets (appelés aussi les nœuds T) et les ceux utilisant l'approche multi-chemins (Les nœuds M). Cela signifie que le réseau est organisé dans les régions en exécutant l'une des deux approches. La difficulté principale est de relier les régions exécutant de différentes structures d'agrégation de données. De cette manière, les règles suivantes doivent être satisfaites [78] :

- *Exactitude d'extremité(Edge Correctness) :* Une extrémité du nœud M ne peut jamais être l'incident à un nœud T. Cela signifie que le résultat d'agrégation dans une région multi-chemins peut seulement être reçu par un nœud M.

- *Exactitude de chemin (Path Correctness*): les nœuds M forment un sous-graphe incluant le puits qui est formé par les arbres composés des nœuds T.

Selon ces règles, le puits est entouré seulement par les nœuds M. Ceux-ci forment une région appelée delta qui peut être étendue ou rétrécie en échangeant les nœuds T en nœuds M et vice-versa, respectivement. Dans la pratique, seuls les nœuds qui se trouvent le long de la limite entre les deux régions changent leur mode d'opération comme le montre la figure V.3 ci-dessous.

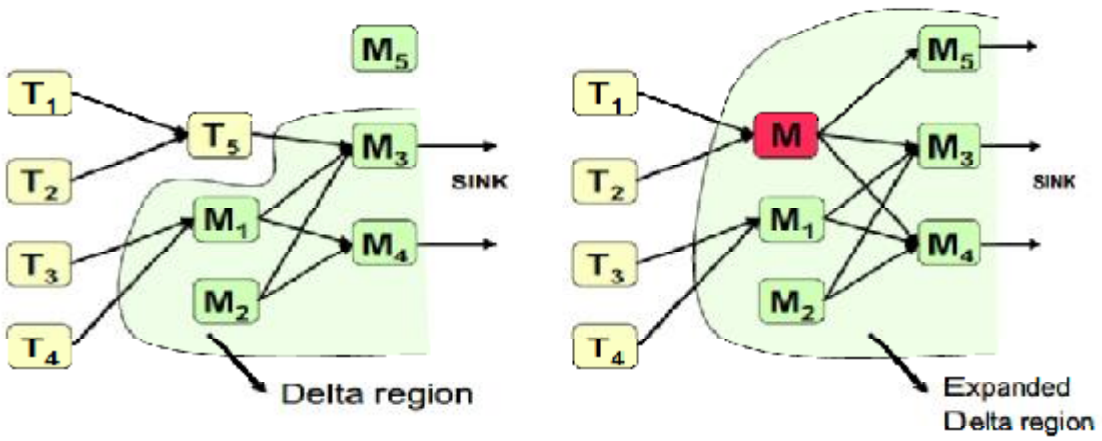


Figure V. 3 Exemple des régions de collecte des données dans Tributaries and Deltas.

V.5.3 La représentation des données [SPZ04, AJA04, ECA03, NCD04]

En raison de ses capacités de stockage limitées, un nœud peut ne pas être capable de conserver tous les paquets reçus et générés dans sa mémoire tampon. Il a besoin de décider donc s'il faut les conserver, les supprimer, les compresser, ou les transmettre. Toutes ces opérations exigent une façon convenable de représenter les données. La structure de données correspondante peut varier selon les exigences de l'application, du nœud ou même de la position du nœud. Ces structures exigent aussi qu'il y ait des fonctions d'agrégation capables de les exploiter. En effet, les fonctions d'agrégation présentées dans les paragraphes précédents comme MAX,MIN,AVG ont un avantage d'être simples et de réduire considérablement la quantité de données à transmette mais ne donnent aucune précision sur les données agrégées(ex :fonctions avec perte).Actuellement, beaucoup de travaux de recherche ont proposé des modèles de représentation de données dans les réseaux de capteurs .Nous avons choisi de détailler *Synopsis Diffusion Framework* proposé dans.

Comme nous l'avons vu dans la section V.5.2.3 de ce chapitre, le protocole Synopsis Diffusion se base sur une approche multi-chemins. La présence des données dupliquées au niveau d'un nœud du réseau faussera donc le résultat de l'agrégation si la fonction d'agrégation utilisée est sensible à la duplication. Les auteurs proposent un ensemble de structure de données et des fonctions d'agrégations pour résoudre ce problème. L'ensemble ainsi formé s'appelle *Synopsis Diffusion Framework.*

Un Synopsis est défini comme un résumé du résultat partiel du processus d'agrégation global reçu à un nœud donné. Trois fonctions sur les synopsis sont possibles pour exécuter l'agrégation de données :

- *Génération d'un synopsis (Synopsis Generation):*Etant donné une lecture d'un capteur, une fonction de génération d'un synopsis SG (.) produit le synopsis correspondant à cette lecture.

- *Fusion des synopsis (Synopsis Fusion) :* Etant donné deux synopsis, une fonction de fusion des synopsis SF (.; .) produit un nouveau synopsis qui résume tous les deux.

- *Evaluation d'un synopsis (Synopsis Evaluation) :* Étant donné synopsis, une fonction d'évaluation d'un synopsis SE (.) donne le résultat final.

V.6 Conclusion

Dans ce chapitre, nous avons étudié les protocoles d'agrégation des données utilisés dans les réseaux de capteurs sans fil. Nous avons constaté que ces protocoles utilisent les fonctions variées allant des fonctions de calcul de la moyenne des données collectées à des fonctions utilisant les requêtes semblables à celles des langages d'interrogation des bases de données.

Dans les applications où l'utilisateur a besoin d'avoir l'information captée par chaque capteur, afin par exemple d'intervenir, les fonctions que nous avons étudiées dans ce chapitre s'avèrent inadaptées. En effet, les fonctions telles que MIN et MAX ne permettent d'avoir qu'une seule donnée à partir des données captées par l'ensemble des nœuds capteurs. L'utilisation des fonctions d'agrégation basées sur les langages d'interrogation des bases de données comme le SQL, peut être couteuse en terme d'exécution et d'espace mémoire utilisé, car dans ce cas la communication doit se faire dans les deux sens (requete-réponse).Il convient donc dans ce genre de situation d'avoir recours à des fonctions utilisant la corrélation entre les données.

Les fonctions d'agrégation de données ont pour but de réduire le nombre de messages transmis dans les RCSFs. Ces messages peuvent causer la congestion et donc la technique d'agrégation des données peut être utilisée pour contrôler la congestion.

Dans la suite de ce travail nous proposons une technique d'agrégation des données pour contrôler congestion dans les RCSFs en se basant sur la corrélation entre les données collectées.

VI.1 Introduction

Dans le quatrième chapitre de ce travail, nous avons présenté et analysé les protocoles de contrôle de congestion proposés dans la littérature. Nous avons remarqué que ces protocoles peuvent ne pas être efficaces dans les applications où il y a beaucoup de redondances des données. Pour remédier à ce problème nous avons vu qu'il est possible d'utiliser les techniques d'agrégation des données. Cela nous a amené encore à étudier les techniques d'agrégation des données dans le cinquième chapitre.

A travers ces deux études nous avons remarqué que le protocole LEACH (Low-Energy Adaptive Clustering Hierarchy) n'est pas conçu pour assurer l'agrégation des données comme un service privilégié, ce qui nous a conduits à proposer une amélioration de ce protocole(LEACH) en utilisant la technique d'agrégation des données.

Dans ce chapitre nous allons décrire la façon dont nous avons implémenté notre nouvelle approche sur le protocole LEACH, puis nous présenterons l'environnement de la simulation en mettant l'axant sur les dispositifs que nous avons utilisé. Nous finirons avec la présentation des résultats d'une analyse comparant les performances de notre implémentation à celle de la version originale du protocole LEACH.

VI.2 Description de la solution

la solution proposé utilise une technique d'agrégation de données pour réduire le nombre de paquets transmis au nœud puits dans le but d'éviter la congestion qui peut être causée par la présence des données redondantes dans le réseau. Cela veut dire quand un capteur *(cluster-heads* CH) recueille une donnée à partir d'un capteur, il ne l'envoie pas directement mais il attend l'arrivée des données venant d'autres capteurs, les agrègent avec sa propre donnée et envoie le résultat d'agrégation au nœud puits. Au moment de cette attente, les données arrivées précédemment doivent être stockées dans une unité de stockage du capteur *(cluster-heads)*.

La communication entre les entités du réseau s'effectue en différentes phases synchronisées au niveau de chaque nœud et la durée de chaque phase dépend de son importance et exigences du protocole.

VI.2.1 Phase de collecte des données

La collecte des données se fait au niveau du nœud puis et au niveau des **cluster-heads.**

1. **au niveau de cluster-head**

Pour accomplir la phase de collecte des données, chaque cluster-head maintient une structure de données (buffer) de taille fixe permettant de stocker les paquets de données (voir tableau) envoyés par les nœuds capteurs. A chaque réception d'une donnée, le capteur calcule le taux d'occupation du buffer. Si ce taux dépasse un certain seuil ou bien atteint son niveau maximal, un capteur commence la phase d'agrégation. Si non il stocke la donnée reçue et la phase continue. Le format d'un paquet de données est décrit dans le tableau ci-dessous.

Champs	Description
sender_id	L'identifiant du nœud émetteur
CH_id	L'identifiant du cluster head
timeStamp	quand le paquet a été envoyé
sensedData	Donnée du phénomène physique étudié
datasize	La taille de donnée

Tableau VI. 3. Les Champs du paquet de données

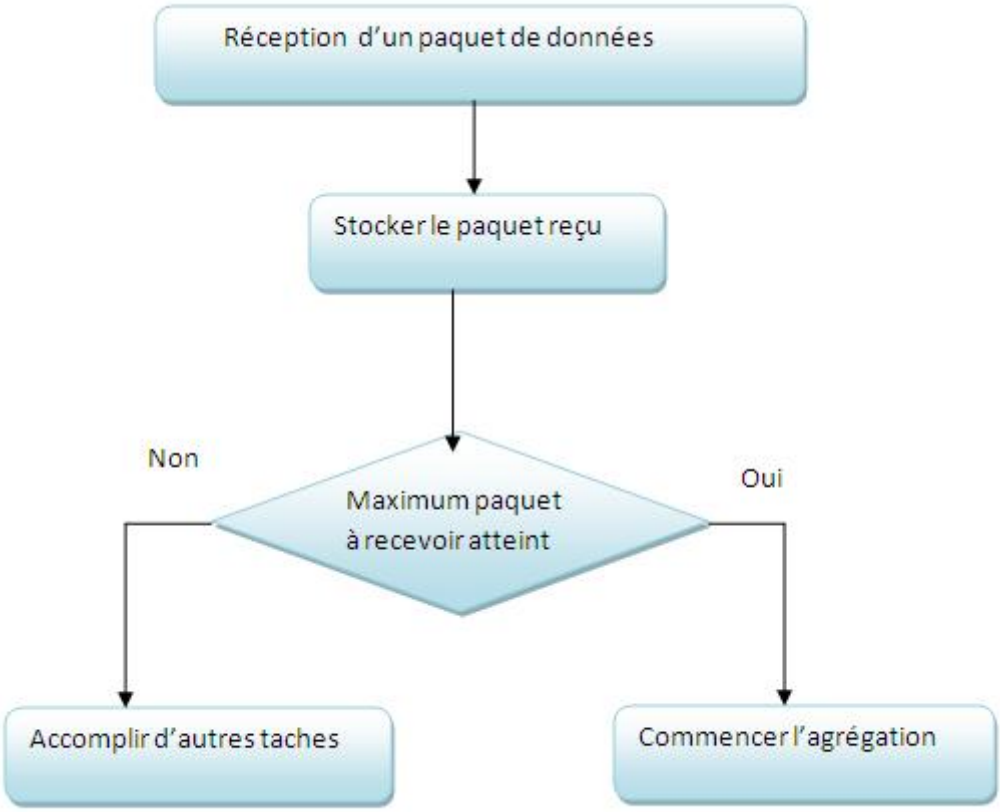


Figure VI. 1. Diagramme de transition d'états de la phase de collecte des données.

2. au niveau du nœud puits

La phase de collecte des données au niveau du nœud puits est très différente de celle qui s'effectue au niveau des CHs. Le puits est un capteur et donc même si nous supposons qu'il possède des ressources suffisantes, ses capacités d'affichage ou d'interprétation des données sont à elles limitées.

En effet, il serait important, pour un utilisateur final, de garder une trace de cette collecte de telle sorte qu'il puisse avoir l'information sur la donnée collectée à chaque moment et par quels nœuds sources ces données ont été collectées. Pour cela nous proposons deux solutions :

a) Après avoir reçu un paquet, le puits le transmet à un ordinateur central et ce dernier stocke l'information utile à l'utilisateur dans un fichier de trace qui est analysé selon les besoins de l'utilisateur est le résultat sera affiché dans une interface graphique ou sur un invité de commande.

b) Le fichier de trace peut être remplacé par une base de données qui sera interrogée selon les besoins de l'utilisateur.

Les fichiers de trace ou la base de données permettront à l'utilisateur de prendre quelques mesures d'intervention. Par exemple soit à une défaillance importante d'un certain nombre de nœuds capteurs, il pourra procéder à un redéploiement d'autres capteurs.

VI.2.2 Phase d'agrégation des données

Pendant la phase d'agrégation des données, la fonction d'agrégation des données est appelée. Cette fonction se base sur la corrélation entre ces données pour former les paquets à envoyer au prochain saut. Pour cela, nous définissons une variable *seuil de sensibilité* (dataSensibility). Tous les paquets de données dont le champ *data* diffèrent d'une valeur inférieure ou égale à *dataSensibility* seront agrégés en un seul paquet dont le champ data est la moyenne arithmétique des données agrégées. Le nombre de paquets à envoyer après l'agrégation dépendra de la corrélation entre les données reçues. Ci-dessous, la fonction utilisée pour assuré l'agrégation de données dans notre solution est illustré :

```
if(Math.abs(((Double)previousData).doubleValue()-
((Double)curentData).doubleValue())<=dataSensibility)
    {
        empiler(positions,(ArrayList)bufferIncData.get(curentData));
        finalData+=((Double)curentData).doubleValue();
        numberOfDataAggregated++;
    }
```

VI.2.3 Phase de contrôle de congestion

Dans le chapitre IV de ce travail, nous avons distingué deux types de congestions à savoir la congestion au niveau du nœud et la congestion au niveau du canal sans fil. Pour contrôler la congestion nous nous sommes basé sur la congestion au niveau de nœud.

Contrôle de congestion au niveau du nœud (cluster-head) **:**

Ce type de contrôle est effectué chaque fois qu'un cluster-head reçoit un paquet de données. Rappelons que le buffer de réception des données a une taille limitée, *bufferSize*, égale au nombre maximum des paquets de données qui peuvent être stockés avant qu'un cluster-head déclare un état de congestion. *BufferSize* est configurable et dépend de l'espace mémoire dont dispose un capteur. Pour vérifier le niveau d'occupation du buffer de données, le CH utilise

un compteur, *bufferOccupied*. A la réception des paquets de données, un nœud capteur vérifie le niveau de congestion comme suit :

Il ajoute à *bufferOccupied* la taille du paquet reçu. Si le pourcentage d'occupation du buffer dépasse un certain seuil, le nœud déclare une congestion à ces nœuds capteurs et commence l'agrégation. Si non il sauvegarde le paquet dans le buffer.

VI.3. Environnement de simulation

Avant sa mise en place, le déploiement d'un réseau de capteurs nécessite une phase de test afin de s'assurer le bon fonctionnement de tous ses composants; matériel ou logiciel. Pour se faire, il existe à ce jour trois grandes solutions **[CRD02]**: le test en environnement réel, la simulation et l'émulation.

1 .Test en environnement réel

La première solution consiste à utiliser un réseau de capteurs réel pour bien mener son évaluation. Bien qu'elle permette d'avoir des conditions parfaitement réalistes, cette solution n'est pas complètement satisfaisante car elle ne permet pas de contrôler l'ensemble des paramètres de l'environnement. De plus, cette solution ne permet pas de reproduire plusieurs fois de suite les mêmes conditions avec précision. En effet, les conditions de propagation du signal à l'intérieur du réseau ne sont pas contrôlables, si bien que d'une expérience à l'autre les conditions observées peuvent évoluer et parfois modifier considérablement le résultat des tests.

2. Simulation

La seconde solution, souvent privilégiée par les chercheurs, est la simulation. Celle-ci permet d'évaluer un modèle d'application ou de protocole dans un environnement contrôlable. Pour cela, la simulation s'appuie sur des modèles décrivant l'environnement, des modèles décrivant les couches de communication utilisées par les terminaux sans fil ainsi que d'autres équipements du réseau et un modèle du trafic circulant sur le réseau. Cependant, la simulation ne travaille pas en temps réel ce qui empêche par exemple l'évaluation d'applications interactives. Utiliser un modèle au lieu de l'implémentation réelle est également pénalisant: la validité du modèle ne garantit pas le bon déroulement de son implémentation et son déploiement. Des erreurs de programmation peuvent toujours survenir au moment de l'implémentation si bien que cette implémentation doit ensuite être évaluée en environnement réel pour vérifier qu'elle se comporte selon son modèle.

3. L'émulation

L'émulation peut être vue comme un compromis entre les deux solutions précédentes en permettant d'évaluer un protocole ou une application dans un environnement contrôlable et reproductible qui simule en temps réel les conditions telles que: les débits, les délais et les pertes que l'on observe dans le réseau cible. Pour cela, l'émulation va simuler les effets des couches basses de telle manière qu'un protocole ou une application s'exécute dans les mêmes conditions que celles de l'environnement réel. Cette solution de test peut par ailleurs être vue comme une étape supplémentaire dans le cycle de développement entre la phase de simulation, qui permet de concevoir les mécanismes spécifiques à un protocole ou à une application et le déploiement dans un environnement réel.

Bien que l'émulation paraisse plus intéressante et porte plus d'avantages, elle impose cependant certaines contraintes. Comme l'émulation travaille en temps réel, les modèles utilisés pour simuler les couches basses ne peuvent pas être trop complexes ce qui implique un impact négatif sur le réalisme de l'émulation rendue. A cet effet, l'approche de simulation détient toujours sa place comme solution pour le test et la validation de protocoles et d'applications **[YJD01]**.

VI.3.1 Le simulateur J-Sim

Plusieurs environnements de simulation sont utilisés pour évaluer les protocoles et des architectures proposés pour les WSNs. Certains sont libres et parfois Open source par contre il y en a d'autres qui sont commercialisés. Aucun de ces simulateurs n'est parfait ni ne répond à tous les besoins. Chacun d'eux présente des avantages et des inconvénients. Dans cette section, nous n'abordons le sujet d'étude comparative de ces outils car cela a fait l'objet de plusieurs travaux de recherche dont **[ASP05]** et **[JPL08]**. Nous allons décrire le simulateur *J-Sim* que nous avons choisi pour simuler notre protocole.

J-Sim **[GOOG]** (appelé autrefois JavaSim, ancien nom qui était en conflit avec la marque de Java Sun) a été développé par une équipe du laboratoire Distributed Realtime Computing Laboratory (DRCL) de l'université d'État d'Ohio. Ce simulateur, développé entièrement en langage Java, repose sur une architecture logicielle basée sur les composants autonomes, appelée « Autonomous Component Architecture » (ACA).

Un composant est une entité indépendante représentant un objet physique (une batterie, un module radio, une couche logicielle, etc.) ou logique (un protocole de routage, un modèle de mobilité, etc.). Ces composants seront ensuite connectés à l'aide de ports afin de générer un réseau simulé.

La simulation du fonctionnement d'un réseau de capteurs, qui exige la définition des composants et leur mise en relation, est réalisée grâce à un langage spécifique, TCL (Tool Command Line) **[TCD11**]. Il s'agit d'un langage de script dans lequel on spécifie

l'architecture du réseau ainsi que les paramètres de simulation et d'analyse. Les commandes de script peuvent également être fournies en ligne de commande, instruction par instruction.

A l'aide de TCL, on définit les composants puis on les connecte. Tous les composants sont hébergés dans un conteneur, qui est à son tour un composant. La définition des composants est en fait la création des objets. Cette création est réalisée par la commande TCL *mkdir*. Chaque composant est d'une entité indépendante (la couche MAC par exemple) qui fonctionne indépendamment des autres entités. Les composants possèdent des ports par défaut pour qu'ils puissent communiquer entre eux. D'autres ports peuvent être créés pour un composant. Une connexion entre deux composants est réalisée par l'intermédiaire de deux ports dédiés, un dans chaque composant (voir la Figure V.4). Cette connexion est matérialisée par la commande TCL *connect*. Il suffit de suivre un schéma qui indique l'interconnexion entre les différents types de composants que l'on veut utiliser. On obtient ainsi l'architecture du nœud. Toujours dans ce script TCL, on y définit les paramètres globaux (par exemple la taille du champ de simulation), les outils de visualisation des résultats et l'ordonnancement de la simulation.

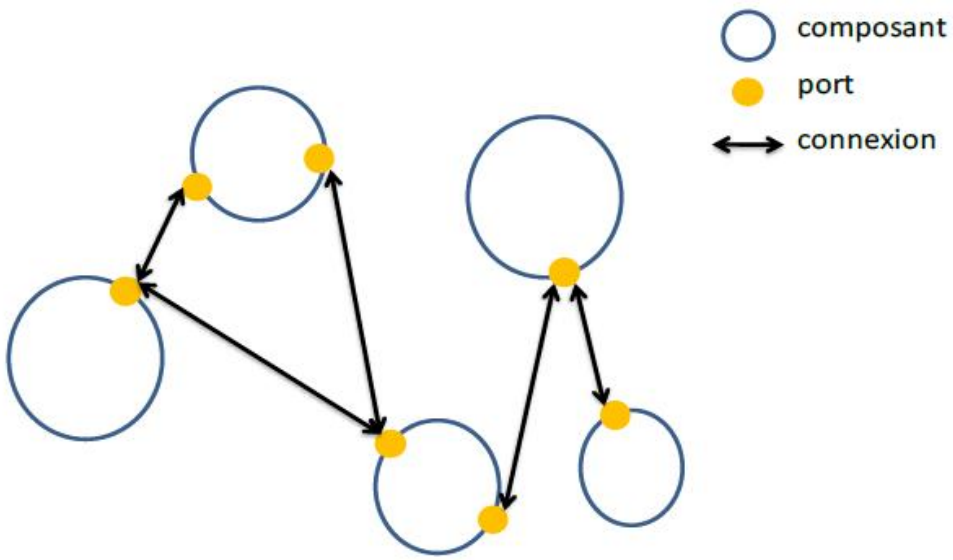


Figure VI. 2. Connexions entre les composants dans J-Sim

On distingue trois types de nœuds dans J-Sim (voir figure V.5): *target*, *sink* et *sensor*. Les nœuds target ont pour fonction de générer des signaux (stimuli) qui peuvent être, par exemple, une secousse sismique, un bruit, etc. Le nœud sink est quant à lui le nœud cible, celui où les informations doivent arriver. Le reste du réseau est formé par les nœuds sensor qui captent le signal généré par target, forment les paquets contenant l'information de mesure et acheminent ces paquets jusqu'au nœud sink.

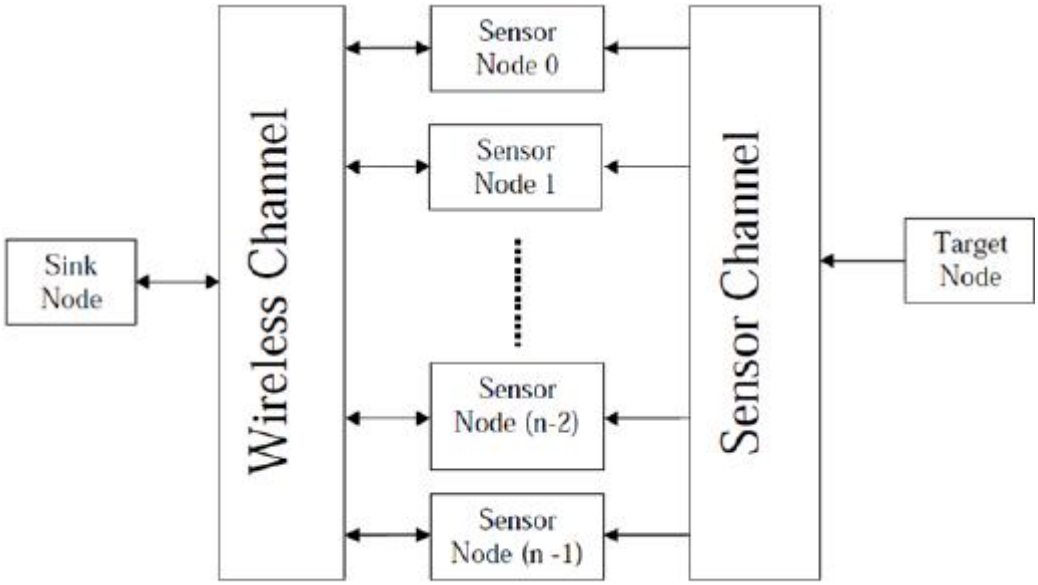


Figure VI. 3. Vue générale sur trois types de nœuds dans J-Sim **[ASJC]**

La Figure **VI. 4.** (partie en pointillé) montre l'architecture interne d'un noeud sensor dans J-Sim. On y distingue notamment les composants CPU et Radio qui définissent les modèles de la consommation énergétique du microcontrôleur et de l'antenne radio. Le composant CPU décrit en principe le coût de traitement des instructions. Le composant Radio décrit le coût de l'envoi et la réception des signaux et la mise en veille. Le composant Battery définit le modèle de la batterie d'un nœud. Ce composant calcule la consommation énergétique due au traitement des instructions, envois et réceptions des signaux en se basant sur les modèles du CPU et de Radio. Un programmeur peut facilement créer ses propres modèles (CPU, Radio, Battery) et les insérer dans un scénario de simulation afin de les tester. A noter que seuls les nœuds sensor sont dotés d'une batterie.

La Figure **VI. 4.** montre aussi que les composants des différentes couches d'un nœud sont séparés (Network layer, Mac Layer, Physical Layer). Ils communiquent entre eux via des ports. Les connexions (désignées par des flèches) entre les composants peuvent être unidirectionnelles ou bidirectionnelles. Une connexion unidirectionnelle d'un composant C1 vers un composant C2 signifie que les informations peuvent être envoyées dans une seule direction (de C1 à C2). D'autre part, une connexion bidirectionnelle signifie un échange d'information dans les deux directions. Le composant Wireless Channel est le composant commun pour tous les nœuds sensor du réseau. Il fait communiquer tous les nœuds sensor entre eux. Chaque nœud sensor du réseau doit avoir une connexion, réalisée par le programmeur, avec ce composant. Le cas échéant, le nœud ne peut pas communiquer avec les autres nœuds. Le composant Sensor Channel est responsable de la communication entre les nœuds target et les nœuds sensor. De même, chaque nœud (sensor ou target) doit avoir une connexion avec ce composant. Si, par exemple, un tel événement est créé par un nœud target,

ce composant envoie l'information à tous les nœuds sensor qui entourent le target (la distribution des nœuds et leurs positions dans le champ de simulation sont définies dans le fichier TCL).

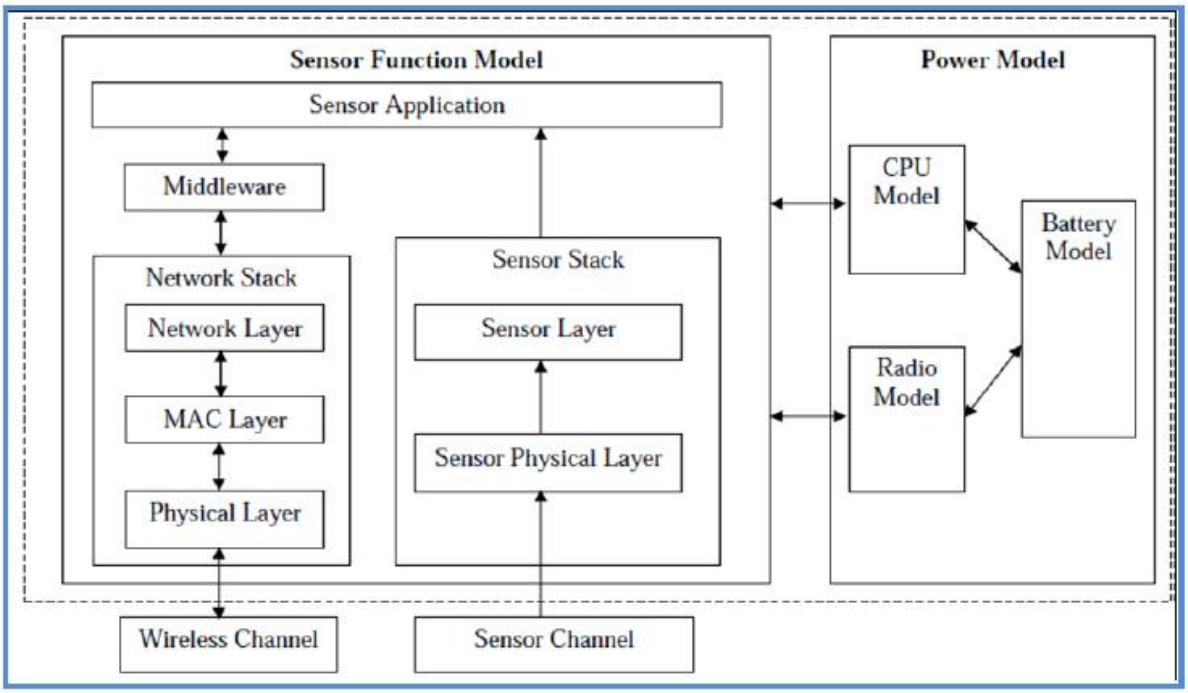


Figure VI. 4. Architecture interne d'un nœud sensor dans J-Sim **[GAD03]**

VI.3.2. L'architecture détaillée de J-Sim [GAD03]

1. Le composant

L'entité de base dans l'architecture logicielle de J-Sim est le composant. Nous définissons une application en réalisant une composition de composants. Les composants dans J-Sim sont faiblement connectés (loosely coupled). Ils communiquent entre eux en connectant leurs ports ensemble, et sont liés par des contrats.

2. Le port

Un composant communique avec les autres composants par l'intermédiaire de ses ports. Il peut posséder plus d'un port. L'interface de programmation entre un composant et son port est bien définie. Un composant peut être développé sans la présence d'autres composants (il se doit d'être autonome). En outre, le mécanisme réel de communication qu'un

composant emploie pour communiquer avec le reste du monde est complètement caché par la notion de port.

3. Le contrat

La communication entre composants est décrite par les contrats. Le contrat impose les conditions sur les ports d'entrée/sortie des composants afin de les faire communiquer. Un contrat indique comment un initiateur (visiteur) et un réacteur (appelé) accomplissent une certaine communication. Il décrit comment un composant répond aux données qui arrivent à chacun de ses ports (par exemple, comment le composant traite les données, certaines structures de données de mises à jour, et produit des sorties à certains ports). Les contrats indiquent la causalité des informations envoyées/reçues entre les composants, mais n'indiquent pas les composants qui participent à la communication.

4. Le langage utilisé pour définir un scénario de simulation

Bien que J-Sim soit écrit en Java, les scénarios de simulation ne sont pas décrits en Java (car ce processus serait trop complexe). La combinaison des langages Java et TCL ne facilite pas cette tâche. Pour développer un grand projet, il peut devenir encombrant d'employer des commandes TCL/Java parce que les références des objets Java doivent être stockées dans des variables TCL afin d'y accéder. Pour simplifier la syntaxe des simulations il existe un système appelé RUntime Virtual (RUV). Ce système s'appuie sur la similitude entre les systèmes composants et les systèmes de fichiers d'UNIX. L'analogie entre un composant/port et un chemin de fichier permet d'accéder au composant de la même manière que l'on accède à un dossier dans un système de fichiers. D'ailleurs des commandes systèmes UNIX, telles que ls, cd, pwd, mkdir, cp, mv, et rm peuvent être utilisées pour manœuvrer des composants et des ports dans le système. Les ports ont une utilisation particulière pour le nommage : a@b (où a est le nom du port et b le groupe du port). En résumant, des composants peuvent être créés et ils peuvent être liés grâce à une interface de commande qui utilise la syntaxe des invités de commandes (shells).

5. Modèle de transmission dans J-Sim

Comme nous pouvons le voir sur la figure VI.5, le nœud target et les nœuds sensors communiquent vivant l'interface appelée sensor channel tandis que la communication entre les nœuds sensors et le sink est réalisée via l'interface Wireless channel. Ces deux interfaces possèdent chacune son propre modèle de transmission.

a) Modèle de transmission dans Sensor channel

La version de J-Sim que nous utilisons (V 1.3), la plus utilisée, implémente deux modèle de propagation du signal généré par le nœud target. Il s'agit du modèle sismique (seismic propagation model) et le modèle acoustique (acoustic propagation model).

Le modèle sismique calcule la puissance du signal reçu(Pr) par un nœud sensor suivant l'équation $Pr=Pt\max(d,d0)fa$

Avec :

P_t : la puissance de transmission du signal

d : la distance entre le target et le sensor,d_0 et

fa :(atenuation factor: facteurs d'atténuation du signal) sont des paramètres configurables du modèle de propagation sismique.

Dans le modèle acoustique, cette formule devient :

$Pr=N(p*\mu g\ ,\sigma g2)$

$p=Ptmax\ (d,d0)fa$, $\mu=U(min_g,max_g)$

ming, maxg, μg et $\sigma g2$ sont respectivement le minimum, maximum, moyen et la variance du gain de microphone et sont configurables.
Notons que les concepteurs de nouveaux protocoles/applications peuvent à tout implémenter leurs propres modèles de transmission ou modifier ces deux modèles.

b) Modèle de transmission dans Wireless channel

Ce modèle de propagation divise le champ de simulation en plusieurs sous-champs à deux dimensions. Chaque sous-champ est un rectangle de taille dx*dy (voir la Figure VI.5 côté gauche). Un nœud définit sa portée de transmission comme suit : il peut communiquer seulement avec les nœuds appartenant à son sous-champ et avec ceux dans les sous-champs voisins. La Figure VI.5 (côté gauche) montre un nœud A et ses nœuds voisins qu'il peut atteindre en un seul saut. Il peut communiquer avec les nœuds appartenant aux neuf sous-champs grisés : le sous-champ où il appartient et les huit sous-champs voisins au sien.

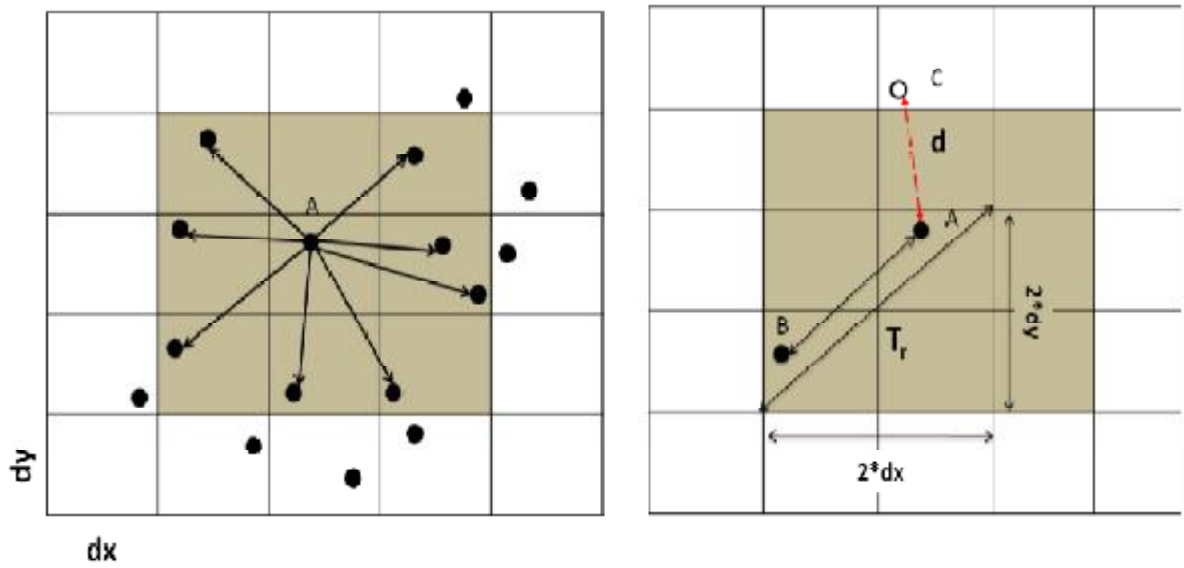


Figure VI.5 Champs de voisinage dans J-Sim

Le rayon maximal de transmission, nommé Tr, est la distance maximale entre deux nœuds communicants. Selon ce modèle de transmission, ce rayon est la diagonale du carré formé par quatre sous-champs (voir la Figure VI.5, côté droit). Ainsi, ce rayon peut être calculé en utilisant la formule $Tr = 2\sqrt{(dx^2+dy^2)}$.

Le côté droit de la Figure VI.5 décrit le comportement de trois nœuds communicants dans J-Sim en se basant sur le modèle de transmission décrit précédemment. Le nœud A ne peut pas communiquer avec le nœud C qui n'appartient à aucun sous-champ des neuf en gris (portée de transmission de A) même si la distance d entre eux est inférieure à Tr. Cependant, le nœud A peut communiquer avec le nœud B qui appartient à un des neuf sous-champs en gris. Ce modèle de propagation est spécifique à J-Sim, qui n'utilise donc ni le modèle théorique sphérique ni celui de voisinage qui en découle et est indépendant d'une distance constante entre l'émetteur et le récepteur. Ceci peut traduire une simulation de l'atténuation du signal à cause des obstacles ou d'autres facteurs (par exemple antenne unidirectionnelle).

VI.4 Simulations et interprétations des résultats

Dans cette partie nous présenterons nos expérimentations, leurs résultats et leurs interprétations. Nous baptiserons A_LEACH la version du protocole LEACH sur la quelle nous avons implémenté notre nouvelle approche.

Dans cette phase, nous allons évaluer le comportement de notre solution en fonction des différents paramètres de configuration pour voir s'il répond bien à notre problématique.

VI.4.1 Paramètres de simulation

Les paramètres pouvant être utilisés pour simuler notre protocole sont nombreux. La plupart d'entre eux sont les paramètres de configuration du simulateur J-Sim. Le tableau ci-dessous donne un certain nombre de paramètres que nous avons utilisés pour évaluer notre protocole.

Zone de déploiement	100 x 100
Nombre de nœuds capteurs	10, 20 ,30 ,40 et50 (stationnaires, déployés aléatoirement)
Nombre de nœuds puits	1 stationnaire
Nombre de nœuds target	5
Sensibilité	2
Rayon de transmission sensor channel	30m
Rayon de transmission wireless channel	30m
Modèle de propagation	acoustique
Temps de simulation	100000s

Tableau V1. 2. Les paramètres de simulation.

VI.4.2 Métriques d'évaluation

Le choix des métriques de performance pour comparer les deux protocoles (LEACH et A_LEACH) dépend de l'objectif de l'évaluation et des relations entre ces métriques. Dans notre cas, Nous avons décidé de nous baser sur le métrique taux d'agrégation, et la durée de vie de réseau.

Au niveau du puits, ce taux est appelé « taux d'agrégation globale » (Tg). Puisque le nœud puits reçoit mais n'envoie pas de paquets de données, le taux d'agrégation globale est défini comme étant le rapport entre le nombre de paquets de données générés par l'ensemble des nœuds (cluster-heads) du réseau sur le nombre de paquets de données reçus par le nœud puits.

Ces taux sont toujours supérieurs à un .Plus le taux d'agrégation globale augmente, plus l'algorithme d'agrégation des données est performant et donc plus on évite la congestion.

$$T_g = \frac{Nombre\ de\ paquets\ de\ données\ générés\ par\ les\ noeuds\ sources}{Nombres\ des\ paquets\ de\ données\ reçus\ par\ le\ noeud\ puits}$$

La durée de vie de réseau est définit comme suit :

La durée de vie de réseau= le temps où tout les nœuds capteur dans le réseau meure.

VI.4.3. Etude de l'impact de la densité du réseau sur les performances des protocoles LEACH et A_LEACH

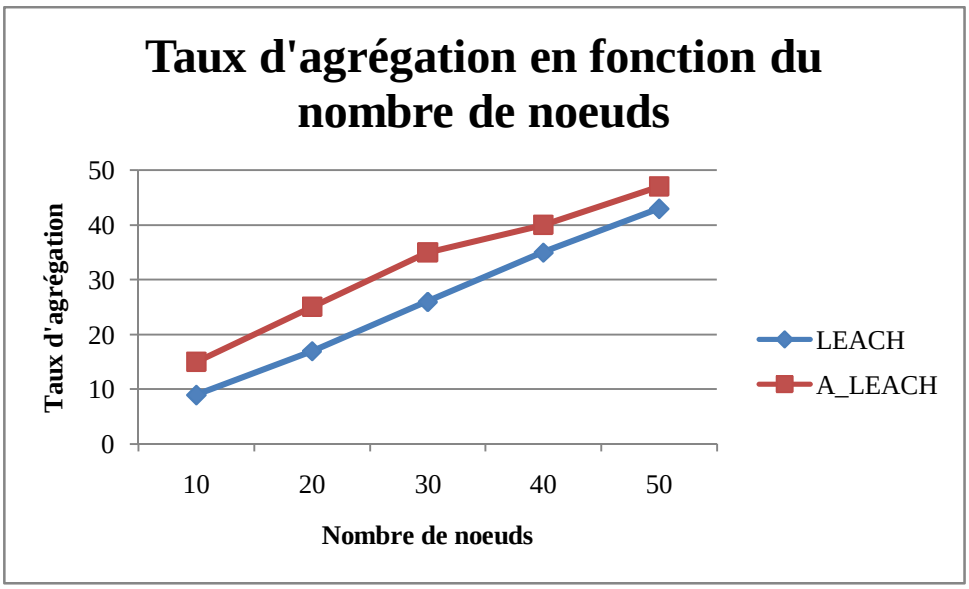


Figure VI.6 Influence de la densité sur le taux d'agrégation

D'après les résultats obtenus nous observons que le taux d'agrégation global est proportionnel au nombre de nœuds du réseau. Les deux protocoles est donc plus performant avec un certain grand nombre de nœuds capteurs, mais le taux d'agrégation dans A_LEACH est supérieur à celui de LEACH est cela est dû a la technique d'agrégation proposé. Cela confirme l'efficacité du mécanisme.

Avec un taux d'agrégation global égal à x, cela signifie qu'en moyenne x paquets produits par l'ensemble des nœuds capteurs (cluster heads) du réseau ont été combinés en un seul paquet. Ce taux varie en fonction de quelques autres paramètres comme la corrélation entre les données du domaine étudié, la taille du buffer de données, etc. En réduisant le nombre de transmission(x augmente), on réduit en même temps le taux d'occupation du média sans fil et donc on réduit la congestion au niveau du canal sans fil.

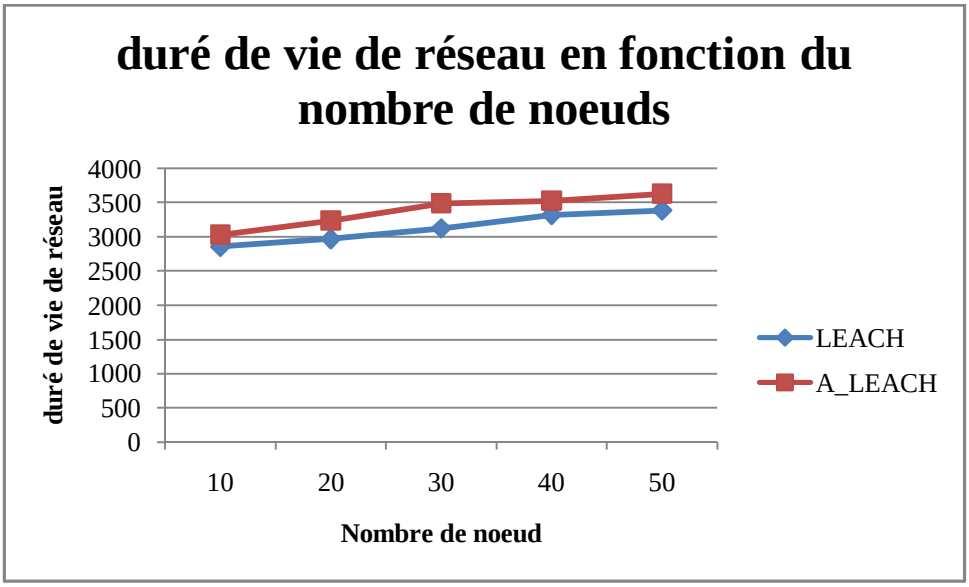


Figure VI.7 Influence de la densité sur la duré de vie de réseau

Comme l'illustre le résultat du graphe **VI.7**, nous remarquons que la duré de vie de réseau est indépendante du nombre de nœuds déployés à cause de la topologie hiérarchique du protocole LEACH qui le rend très scalable. En effet, quand la taille du réseau augmente, le nombre de CH augmente. Donc, les nouveaux nœuds vont être affectés aux nouveaux CH et regroupés indépendamment des groupes déjà existants dans le réseau. Donc, malgré l'augmentation du nombre de nœuds déployés, la taille de tous les groupes est la même. Ainsi, tous les CH effectuent le même taux de tâches. Ainsi, LEACH maintient la consommation d'énergie des nœuds quelque soit la taille du réseau.

Par ailleurs, nous constatons que A_LEACH maximise la durée de vie de réseau par rapport a LEACH ; ceci explique l'efficacité de la nouvelle technique pour l'agrégation des données, qui minimise au maximum le nombre des message transmise puisque les modules radio des nœuds capteur vont passer la plus part de temps dans l'état d'attente (idle state), qui consomme moins d'énergie par rapport à l'état de réception (recepting state) et encore moins par rapport à l'état de transmission (transmitting state).

VI.5. Conclusion

Dans ce chapitre, nous avons présenté l'implémentation ainsi que l'évaluation des deux protocoles LEACH et A_LEACH. Le simulateur J-Sim est utilisé. Il consiste une programmation entière en langage java. Nous nous sommes intéressées aux métriques de performance qui est le taux d'agrégation et la durée de vie de réseau.

Les simulations ont été menées sous des conditions différentes de densité et de temps et dans tout les cas de figure A_LEACH a surperformé LEACH

Conclusion générale

Les RCSF vont sans doute dans les années à venir constituer un développement technologique majeur apportant des solutions aux différents problèmes dans plusieurs domaines d'applications liés à la sécurité, la santé, l'agronomie, la domotique, etc.

Cependant, il reste encore de nombreux problèmes à résoudre dans ce domaine afin de pouvoir les utiliser dans les conditions réelles. L'un des problèmes qu'on peut rencontrer dans ce genre de réseau est limitation de l'énergie et la congestion causée par les données redondantes. Cette congestion est liée à la nature des réseaux de capteurs, principalement leur mode de déploiement. Quand un réseau est congesté, il y a un gaspillage des ressources dont la bande passante.

Dans ce projet, nous avons mis en place une technique d'agrégation des données dans le protocole de routage LEACH. Il consiste à combiner les données redondantes ou fortement corrélées afin de réduire le nombre de messages transmis par les capteurs et donc pouvoir contrôler la congestion et économisé de l'énergie au niveau de la couche réseau.

A travers l'évaluation des deux protocoles (LEACH et A_LEACH), nous avons constaté que les résultats de la solution proposée (A_LEACH) sont satisfaisants en terme de taux d'agrégation et la durée de vie de réseau essentiellement a forte densité de nœud et qu'ils surperforment ceux de LEACH en toutes circonstance.

Comme perspectives de notre travail, nous allons continuer à l'améliorer et surtout en prenant en compte d'autres métriques comme le taux de perte des paquets, le niveau de congestion au niveau de chaque nœud du réseau, sa scalabilité, l'intégration d'une base de données au niveau de l'ordinateur central pour collecter les données et le comparer avec d'autres protocoles. Nous envisageons tous cela dans nos futurs travaux de recherche qu'ils soient d'ordre individuels, professionnels ou académiques.

Références bibliographiques

A

[ATA07] Amhammedi Ahmed, Targui Mohammed, **«Conception et réalisation d'unsimulateur pour la Surveillance d'un Réseau de Capteurs sans fil »,** Projet find'étude, Institut des sciences exactes, Département d'informatique, CentreUniversitaire de Bechar, Juin 2007.**[ISA07a]**

[AAJ07] AbdallahMakhoul, Ahmed Mostefaoui, Jacques Bahi, **« A Mobile Beacon BasedApproach for Sensor Network Localization »**, Third IEEE International ConferenceonWireless and Mobile Computing, Networking and Communications, Page(s): 44 –44, Washington DC, USA, 2007.

[ABD07] Abdelhakim Hamzi, **« Plateforme basée agents pour l'aide à la conception et lasimulation des réseaux de capteurs sans fil** », Mémoire de magister, InstitutNational de formation en Informatique (I.N.I), Algérie, 2007.

[AHY08] Abdelmadjid Bouabdallah, Hatem Betthahar, Yacine Challal, « **Les Réseaux decapteurs (WSN: Wireless Sensor Networks)** », Cours, Université de Technologie deCompiègne, France, 2008.

[AJA04] Ahmed Kamal, Jamal Al-Karaki, **« Routing Techniques in Wireless SensorNetworks: A Survey** », IEEE Wireless communications, Page (s): 6-28, Iowa StateUniversity, USA, 2004.

[AKK05]K.Akkaya, M.Younis, "A Survey on Routing Protocols for Wireless Sensor Networks", Elsevier ad hoc network journal, No 3, 2005.

[AKY02]L.Akyildiz, W.Su, Y.Sankarasubramaniam, E.Cayirci, "A survey on sensor networks", IEEE communication magazine, vol 40, No 8, August 2002.

[ADJ05] AdjidoIdkiwa**, « La sécurité des réseaux wi-fi»,** Master 1, Faculté de Nantes,France, 2004/2005.

[AJO05]A. D. JOSEPH, « Energy harvesting projects », Pervasive Computing, vol. 4, pp. 69–71, 2005, IEEE.

[ACC07]Aldar CHAN et **Claude CASTELLUCCIA**, « On the security of concealed data aggregation », dans le Colloque européen sur la recherche dans le domaine de la sécurité d'ordinateur (ESORICS 2007), Septembre 2007, Dresden, Allemagne.

[ASJC]Ahmed Sobeih et Jennifer C. Hou, « A simulation framework for sensor networks in j -sim »

B

[BOU07] BouabdellahKechar**, « Problématique de la consommation de l'énergie dans lesréseaux de capteurs sans fil »,** Séminaire LIUPPA, Université de Pau et des Pays del'Adour, 14 Octobre 2007.

[BFL03] Bruno Tharon, Francis Dupont, LoutfiNuaymi, Sylvain Gombault, Valérie Gayraud,**«La Sécurité dans les Réseaux Sans Fil Ad Hoc»,** Conférence SSTIC03, France, 12Juin 2003.

[BHA05]Bhaskar KRISHNAMACHARI, «Networking Wireless Sensors », Cambridge University Press, 2005.

C

[CRO08] Crossbow Technology-logo, **« Wireless measurement system mica2 »,** Revue, www.xbow.com, 2008.

[CHM07] Cedric Richard, Hichem Snoussi, Mehdi Essoloh, **« Localisation distribuée dans lesréseaux de capteurs sans fil par résolution d'un problème quadratique »**, Revue,éditée par : GRETSI, Groupe d'Etudes du Traitement du Signal et des Images,Université de Technologie de Troyes, Septembre 2007.

[CSA03]C.Y. WAN, S. B. EISENMAN et **A. T. CAMPBELL**, « CODA: Congestion Detection and Avoidance in Sensor Networks » démarches de ACM SenSys, Novembre 2003.

[CRB04]C.-T. EE et**R. BAJCSY**, « Congestion Control and Fairness for Many-to-One Routing in Sensor Networks », démarches de ACM SenSys, November 2004.

[CAP03]C. HAOWEN et**A. PERRIG**, « Security and privacy in sensor networks », Computer, vol. 36, pp. 103–105, 2003.

D

[DJP03] David Culler, Joe Polastre, Phil Levis, Rob Szewczyk, **«Building Sensor Networkswith TinyOS »**, The International Conference on Mobile Systems, Applications, andServices 2003(mobysys03), California University, UC Berkeley, 5 May 2003.

[DJA08] Djallel Eddine Boubiche, **«Protocole de routage pour les réseaux de capteurs sansfil»**, Mémoire de magistère, Université de l'Hadj Lakhdar, Batna, Algérie, 2008.

[DUT07] rapport réalisé par *DUTREIGE Jonathan* et *TIMMERMANS Thomas*.Master professionnel S .I.R. Année 2006-2007

E

[EYA07] EyaDhib, « **Routage avec QoS temps réel dans les réseaux de capteurs »,** Projet find'étude ingénierie de réseaux, Ecole Supérieure des Communications de Tunis, 2007.

[ACC07]Aldar CHAN et **Claude CASTELLUCCIA**, « On the security of concealed data aggregation », dans le Colloque européen sur la recherche dans le domaine de la sécurité d'ordinateur (ESORICS 2007), Septembre 2007, Dresden, Allemagne.

F

[FAT07] Fatima Zohra Benhamida, **« La tolérance aux pannes dans les réseaux de capteurssans fil** », Rapport du mini projet, Institut National de Formation en Informatique INI,Algérie, 2006/2007.

G

[GSO02] Garcia Luna Aceves, Soumya Roy, **« Node-Centric Hybrid Routing for Ad HocNetworks** », mobiwac, International Mobility and Wireless Access Workshop(MobiWac'02), Page(s): 63-63, University of California at Santa Cruz, 2002.

[GBI00] G. Bianchi «Performance analysis of the IEEE 802.11 distributed coordination function»,IEEE Journal on Selected Areas in Communications, March 2000.

[GAD03]Gianni A. Di Caro, « Analysis of simulation environments for mobile ad hoc Networks »,Technical Report No. IDSIA-24-03, DalleMolle Institute for Artificial Intelligence, Galleria 2, 6928 Manno, Switzerland, 2003.

[GOOG] http://sites.google.com/site/jsimofficial/.

H

[HNA06] Hani Hadjammar, NaouelDoufene, **« Routage dans les réseaux de capteurs :optimisation du protocole Directed Diffusion»,** Projet de fin d'étude, InstitutNational de formation en Informatique INI, Algérie, 2006.

I

[IMO06] ImadMahgoub, Mohammad Ilyas, **« Sensor Network Protocol** », Hardcover Book,ISBN: 0849370361, Number of pages: 248, USA, 27 Janvier 2006.

[ISA07a] Isabelle GuerinLassous, **« Réseaux Ad Hoc, réseaux de capteurs »**, Cours M2Recherche RTS, RTS5, Page(s) : 05-10, Université de Lyon, 26 Septembre 2007.

[ISA07b] Isabelle Guérin Lassous, **« AutonomicComputing : Accès au médium radio »**, Cours M2 Recherche RTS, RTS5, Page(s) : 43-95, Université de Lyon, 15 Septembre 2007.

[IKO04]I. SOLIS et **K. OBRACZKA**, « The Impact of Timing in Data Aggregation for Sensor Networks » ,dans IEEE ICC 2004, Juin 2004, Paris, France.

J

[JKR04] J. Hart, K. Martinez, R. Ong, **« Glacsweb: a sensor network for hostileenvironments »,**First IEEE Communications Society Conference on Sensor and AdHoc Communications and Networks, Santa Clara, USA (in press), October 2004.

[JPL08]Johannes Lessmann, Peter Janacik, Lazar Lachev et DalimirOrfanus, « Comparative Study of Wireless Network Simulators»,Seventh International Conference on Networking,Octorber 2008.

K

[KEL08] www.Kelkoo.fr ,**« Cle USB Bluetooth pc », 2008.**

[KDT07] K. SOHRABY, D. MINOLI, T. ZNATI, Livre :“ WIRELESS SENSOR NETWORKS

[KRI02]B.Krishnamachari, D.Estrin, S.Wicker, "Modeling Data Centric Routing in Wireless Sensor Networks", Proceeding of 7thAnnual ACM/IEEE INFOCOM, New York, June 2002.

[KKN11]Kiran MARAIYA, Kamal KANT et **Nitin GUPTA**, « Wireless Sensor Network: A Review on Data Aggregation », revue internationale de la recherche scientifique et technologique, Vol. 2, N°4, Avril 2011.

[KMY05]K. AKKAYA et M. YOUNIS, « A Survey of Routing protocols in wireless Sensor networks », revue Elsevier Ad Hoc Network, vol. 3, N° 3, pp. 325–349, Mai 2005.

L

[LNA04] Lyes Khelladi, NadjibBadache« **Les réseaux de capteurs: état de l'art** », Rapportde recherche, Algérie, Février 2004.

[LNA08] LyesKhelladi, NadjibBadache, « **Improving Directed Diffusion With Power-Aware Topology Control For Adaptation to High Density** », LOCALGOS'08workshop, in conjunction with The 4th IEEE/ACM International Conference onDistributed Computing In Sensor SystemS (DCOSS 2008), Algeria, 2008.

[LNA04] Lyes Khelladi, NadjibBadache« **Les réseaux de capteurs: état de l'art** », Rapportde recherche, Algérie, Février 2004.

M

[MWI06] Mathieu Badnet, Nicolas Belloir**«Réseaux de capteurs : Mise en place d'uneplateforme de test et d'expérimentation** », Master Technologie de l'Internet 1ère année, France, 2005/2006.

[MOU05] Mounir Achir, **« Technologies basse consommation pour les réseaux Ad Hoc**», Thèse, Institut National Polytechnique de Grenoble, 06 Juillet 2005.

P

[PSY08] Prométhée Spathis, Serge Fdida, YosraBarouni, « **Modèle générique pour le routage orienté contenu** », Document scientifique, hal-00260342, Laboratoired'informatique de Paris 6, Université Pierre et Marie Curie, Mars 2008.

[POT00]G.J.Pottie, W.J.Kaiser, "Wireless Integrated Network Sensors", Commun.ACM, vol. 43, no. 5, May 2000.

[PRE05] PreethaRadhakrishnan, **"Enhanced routing protocol for graceful degradation in wireless sensor networks during attacks"**, Thèsed'ingéniorat, Université deMadras, Chennai, Décembre 2005.

R

[RRA08] Réseaux Radioélectriques, **« Croissance des services radioélectriques en Octobre2008 »**, Magasine, Stratégies Télécom &Multemedia, France, 30 Octobre 2008**.**

[RRV07] Rampur Srinath, R. Srinivasan, Vasudev Reddy, **« AC: Cluster Based SecureRouting Protocol for WSN »**, IEEE Computer Society, Third InternationalConference on Networking and Services (ICNS'07), Page(s): 44-45, 19-25 June 2007.

[RCK10]Rekha CHAKRAVARTHI, C. GOMATHY, Suraj K. SEBASTIAN, K. PUSHPARAJ et **VinuBinto MON**, «A Survey on Congestion Control in Wireless Sensor Networks», Revue internationale de l'informatique et de la communication, Vol. 1, N°1, pp. 161-164, Janvier-Juin 2010.

[RTHCC]R.THEN MALAR « Congestion Control in Wireless Sensor Networks Based Multi-Path Routing In Priority Rate Adjustment Technique»

[REJ04]R. SZEWCZYK, E. OSTERWEIL, J. POLASTRE, M. HAMILTON, A. MAINWARING et**D. ESTRIN**, « Habitat monitoring with sensor networks », Communications de l'ACM, vol. 47, pp. 34–40, 2004.

S

[SEV06] Séverine Sentilles, **« Architecture logicielle pour capteurs sans-fil en réseau »,**Master TI 2e année, MälardalenUniversity, Sweden, Janvier-Juin 2006.

[STE04] Sébastien Tixeuil, Ted Herman, **« Un algorithme TDMA réparti pour les réseaux de capteurs »**, INRIA Projet Grand Large, Universités Iowa et Paris-Sud XI, 2004.

[SCK02]S. LINDSEY, C. RAGHAVENDRA et K. M. SIVALINGAM, « Data Gathering Algorithms in Sensor Networks using Energy Metrics », IEEE Trans. Parallel Distrib. Syst., vol. 13, N°.9, pp. 924–935, Septembre 2002.

[SPZ04]S. NATH, P. B. GIBBONS, Z. R. ANDERSON et **S. SESHAN**, « Synopsis Diffusion for Robust Aggregation in Sensor Networks », dans ACM SenSys 2004, Novembre 2004, Baltimore, MD, US.

[SMJ02]S. MADDEN, M. J. FRANKLIN, J. M. HELLERSTEIN et W. HONG, «TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks», dans OSDI 2002, Décembre 2002 Boston, MA, U.

T

[TAY00] TayebLemlouma, « **Le routage dans les réseaux mobiles Ad Hoc »,** Mini projet,Institut National de Recherche en Informatique et Automatique INRIA, 2000.

[TEC08] www.techno-science.net, 2008.

[TCD11]TclDeveloppper Site. [En ligne] Mai 2011. http://www.tcl.tk/.

[TJS05]T. ARAMPATZIS , J. LYGEROS et **S. MANESIS**, « A Survey Of Applications OfWirelessSensors And Wireless Sensor Networks », dans la conference méditerranéenne sur la commande et l'automatisation MED05, 2005, Nicosia, Cyprus.

V

[VIO08] Violeta Felea, **« Routage dans les réseaux de capteurs sans fil »,** Journées ResComStrasbourg, Université de franche.comté, 9-10 Octobre 2008.

W

[WEN00] Wendi Beth Heinzelman, **«Application-Specific Protocol Architectures for Wireless Network** », IEEE Transactions on Wireless Communications,Massachusetts Institute of Technology, June 2000.

X

[XZK06]X. JIANBO, Z. SILIANG et **Q. FENGJIAO**, « A new In-network data Aggregation technology of wireless sensor networks », Dans IEEE SKG'06, Novembre 2006, Guilin, China

Y

[YAS06] Yasmina Khalfaoui, **«Routage dans les réseaux de capteur sans fils»,** Projet de find'étude, Centre universitaire Mustapha Stambouli, Mascara, 2006.

[YAS07] Yasser Romdhane, **« Evaluation des performances des protocoles S-MAC etDirected Diffusion dans les réseaux de capteurs** », Projet de fin d'études, EcoleSupérieure des Communications de Tunis (Sup'Com), 2006 / 2007.

Z

[ZSJ02]Z. FENG, S. JAEWON et **J. REICH**, « Information-drivendynamicsensor collaboration », Magazine: Signal Processing, vol. 19, pp. 61–72, 2002, IEEE.

I Installation du simulateur JSIM sous la plateforme WINDOWS

I.1 Préparation d'environnement

Etape1 :

Pour installer le simulateur JSIM, il faut déjà avoir installé sur votre PC un logiciel spécial depuis le site web de la société Sun Microsystems, son nom complet est ***Java 2 Software Development Kit (J2SDK)***. Pour des raisons de stabilité, il est hautement recommandé d'installer la version ***J2SDK 1.4*** ou bien des versions ultérieures. La version dont j'ai utilisé pendant mon projet de PFE est ***J2SDK 5.0*** et peut être téléchargée depuis le site suivant :
http://java.sun.com/j2se

Etape2 :

La compilation des codes sources pour le simulateur JSIM peut se faire soit via le compilateur *Apache Ant*, soit par la commande « *make* ».
Le premier compilateur est valable depuis le lien : http://ant.apache.org/
Pour compiler via la commande « *make* », on doit installer l'utilitaire *GNU-Make makefiles* in Java. Cet utilitaire permet d'installer des commandes propres aux plateformes LINUX sous WINDOWS.

Etape3 :

La version du simulateur JSIM la plus utilisé est J-SIM version1.3. Elle peut être téléchargée depuis le lien : http://www.j-sim.org/cgi-bin/j-sim_downloader.cgi Il faut noter que la version J-Sim version1.3 n'est pas compatible avec la version **J2SDK 5.0**.
Pour des questions d'interopérabilités, on doit installer le patch4 ou bien des versions de patch ultérieures. Le patch dont j'ai installé est ***patch4-version1.3*** valable sur: http://www.jsim.org/patch.html

I.2 Installation de JSIM

Etape4 :

L'installation du JSIM se fait sur plusieurs sous étapes:

On décompresse sous la racine C : \\ le package téléchargé J-Sim_v1.3.tar. On obtient par la suite un dossier nommé jsim-1.3.
Rem : la décompression des fichiers .tar sous WINDOWS se fait avec le logiciel WINRAR.
On modifie le fichier setcpath.bat placé sous C:\\jsim-1.3 comme suit :
On affecte à la variable **%J_SIM%** le chemin d'installation du simulateur JSIM

On affecte à la variable **% JAVA_HOME %** le chemin d'installation de **J2SDK5.0**
On initialise une variable **% ANT_HOME %** au chemin d'installation du compilateur *Apache Ant* et on ajoute à la variable **%PATH%** le chemin vers les commandes de compilation si on le choisit comme compilateur pour JSIM.
Si on a choisit de compiler les codes sources du simulateur JSIM avec la commande « *make* », on doit donc ajouter le chemin d'installation de à la variable **%PATH%**.
On ajoute à la variable **%CLASSPATH%** le chemin vers les classes du simulateur JSIM.
Si le compilateur choisi pour le simulateur JSIM, est *Apache Ant*, le fichier *setcpath.bat* doit comprendre les données suivantes :

```
@echo off
set JAVA_HOME=C:\jdk
set J_SIM=C:\jsim-1.3
set ANT_HOME=C:\ant
set PATH=%PATH%;c:\jdk\bin;c:\ant\bin
set CLASSPATH=.;c:\jsim-1.3\classes;c:\jsim-1.3\jars/tcl.zip;c:\jsim-
    1.3\jars/jython.jar
```

Si la compilation pour le simulateur JSIM est suivant la commande « *make* », le fichier setcpath.bat doit être de la forme suivante :

```
@echo off
set JAVA_HOME=C:\jdk
set J_SIM=C:\jsim-1.3
set PATH=%PATH%;c:\jdk\bin;c:\UnxUtils_1\usr\local\wbin
set CLASSPATH=.;c:\jsim-1.3\classes;c:\jsim-1.3\jars/tcl.zip;c:\jsim-
    1.3\jars/jython.jar
```

1. On ouvre maintenant la fenêtre des invites des commandes DOS, et on se place sous le répertoire jsim-1.3. Pour cela, on tape les commandes suivantes :

```
cd C:\\
cd jsim-1.3
```

2. On exécute maintenant le fichier setcpath.bat en l'appelant sous la fenêtre d'invite des commandes.

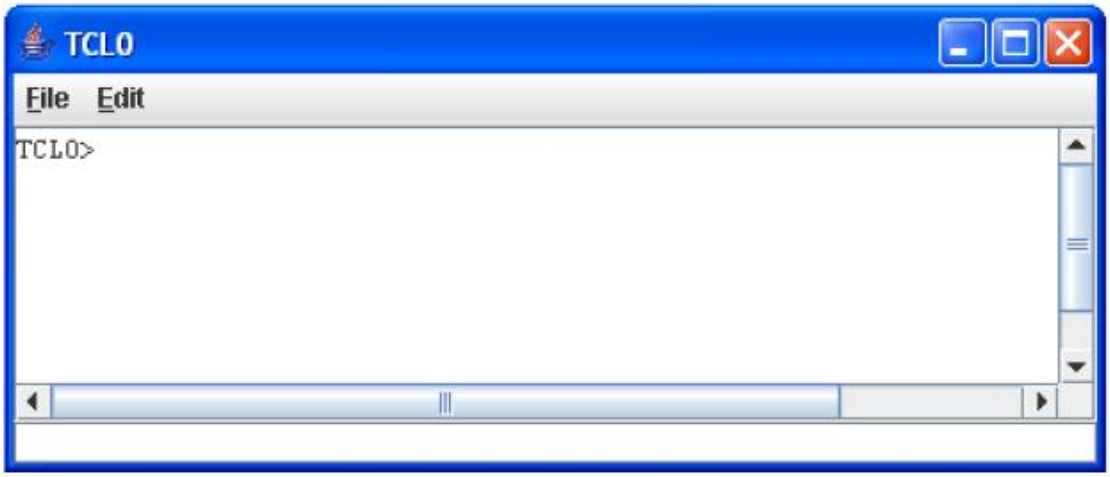
3. On compile tous les codes sources du simulateur JSIM en tapant la commande « *make* » si vous avez installé l'utilitaire *GNU-Make makefiles in Java* ou la commande « ant compile » si vous avez installé le compilateur *Apache Ant*.

4. On décompresse maintenant le dossier ***patch1.3-4.tar.gz***, déjà téléchargé, sous le répertoire jsim-1.3 et on accepte les mises à jour des fichiers existants.

5. On recompile de nouveau les codes sources et on ignore tous les avertissements.

Etape5 :

Pour vérifier si l'installation de JSIM a réussi, vous devez avoir une fenêtre pop up intitulé TCL0 qui apparait en tapant la commande suivante : java drcl.ruv.System



Etape 6 :

Chaque fois que vous désirez simuler un script TCL, vous devez changer le chemin courant vers où est placé votre script, puis vous tapez la commande suivante :
java drcl.ruv.System « nom de votre script.tcl »

II Intégration du code source du protocole dans J-Sim

Le code source du protocole à simuler, qui doit être écrit en langage Java est copié dans l'un des sous-répertoires du répertoire */src* de J-Sim, à condition de respecter le principe de développement imposé par le langage Java (notion de packages).

Pour faciliter le développement du code source, il est nécessaire de choisir l'un des environnements de développements pour le langage Java. Il en existe plusieurs mais les plus populaires sont Eclipse IDE [et NetBean IDE .Nous avons choisi NetBeans IDE pour sa facilité à

créer les projets qui nous a permis d'ailleurs d'installer J-Sim dans le répertoire */pro* de NetBeans IDE.

L'ensemble de J-Sim et NetBeans combinés, nous procédons comme suit pour simuler notre protocole :

Nous créons un projet sous NetBeans dans lequel nous installons J-Sim. Une fois le code source est sauvegardé, sous netBeans, nous le compilons par la commande « *ant compile »* (voir guide d'installation de J-Sim cité précédemment).

III Définition de quelques méthodes du protocole

Parmi les méthodes de traitement de notre protocole nous pouvons donner :

- *protected void _start ()* : exécutée au démarrage d'un nœud.
- *protected void _stop()*: exécutée à l'arrêt d'un nœud .
- *protected synchronized void recvSensorEvent(Object data_)* :exécutée à la réception d'un stimulis généré par un nœud target.
- *protected synchronized void recvSensorPacket(Object data_)*:exécutée quand un nœud reçoit un paquet.
- *protected synchronized void recvDATA(LEACH_Data_Packet msg)* exécutée quand un nœud (cluster head) reçoit un paquet.
- *public synchronized void sendData():* exécutée par un nœud capteur pour envoyer un paquet.
- *public synchronized void sendCongNotifPkt()*: exécutée par un noeud pour envoyer un paquet de notification de congestion.
- *public synchronized void hdlIncCongNotifPkt(CongNotifPkt congPkt_):*exécutée par un nœud lorsqu'il reçoit un paquet de notification de congestion.
- *public synchronized void processAggregation()*: pour effectuer l'agrégation
- *public synchronized void processOther(Object data_, Port inPort_)*: exécutée quand un composant reçoit un message via un port *inPort_*
- *protected synchronized void timeout(Object data_)*:exécutée par un composant à l'expiration d'un timer relatif à un événement.
- *public synchronized void SendDataToBS()* exécutée par un nœud (cluster head) pour envoyer un paquet a la station de base.

- public synchronized void insertMyData() Ajouter la donnée captée ds le buffer avant l'aggrégation'

IV Exécution et simulation

Pour lancer la simulation, nous avons deux possibilités :

Charger le fichier de configuration de notre protocole via l'interface RUV (voir paragraphe précédent),ou utiliser la commande *« java drcl.ruv.System –u nom_fichier_de_configuration ».*

Le nom du fichier de configuration est de format **.tcl*