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

**Université de Batna 2**
**Faculté de Mathématiques et D'Informatique**

**Département d'Informatique**

# Thèse

*En vue de l'obtention du diplôme de*
**Doctorat en Informatique**

---

# UN PROTOCOLE DE ROUTAGE TOLERANT AUX PANNES POUR LES RCSF

---

*Présentée Par*
**Djebaili Yasmine**

Soutenue le : 20/06/2018

## Membres du jury :

| | | | |
|---|---|---|---|
| ***Président*** | *TALHI Said* | *MCA* | *Université de Batna 2* |
| ***Rapporteur*** | *BILAMI Azeddine* | *Professeur* | *Université de Batna 2* |
| ***Examinateurs*** | *CHAOUI Allaoua* | *Professeur* | *Université de Constantine 2* |
| | *BOUAM Souheila* | *MCA* | *Université de Batna 2* |
| | *MERAH Elkamel* | *MCA* | *Université Abbess Laghrour, Khenchela* |

# REMERCIEMENTS

*« Nous sommes des nains juchés sur les épaules de géants »*
***Isaac* Newton** *(Proverbe).*

*Je tiens premièrement à me prosterner remerciant mon Dieu ALLAH le tout puissant de m'avoir donné la force morale et physique pour achever cette thèse.*

*Tout d'abord, je remercie le Professeur Bilami Azzedine, pour m'avoir honoré par son encadrement, sa disponibilité indéfectible, ses conseils précieux, ses nobles valeurs humaines pendant les moments difficiles de ma vie de doctorante ainsi que pour la confiance qu'il m'a témoignée jusqu'à l'aboutissement de ce travail. Avec patience et pédagogie, il m'a fait découvrir plusieurs facettes de l'activité de chercheur.*

*Mes remerciements vont également aux membres de jury qui me font le grand honneur de juger mon travail : Je remercie Docteur Talhi Said qui a bien voulu présider le jury. Je remercie également les examinateurs : Professeur Chaoui Allaoua, Docteur Merah Elkamel et Docteur Bouam Souheila pour l'intérêt qu'ils ont porté à mon travail.*

*Un grand merci va également à l'ensemble des membres du laboratoire LaSTIC, avec lesquelles j'ai pu avoir de nombreux échanges scientifiques.*

*Je tiens aussi à exprimer toute ma gratitude à tous les enseignants du département d'informatique qui ont contribué à ma formation et pour leur soutien moral et leurs encouragements, un grand merci à toutes les personnes qui, de près ou de loin, m'ont apporté leur soutien technique, scientifique ou moral pendant les moments difficiles.*

*Je ne pourrais oublier mes amies Amal et Houda que je remercie pour leurs encouragements permanents dans les moments difficiles durant mon doctorat.*

*Bien sûr, je remercie mes parents, qui avec leur amour, leur esprit de sacrifice et leur soutien sans fin, m'avaient toujours guidé et encouragé, et qui ont tout fait pour que je puisse réussir dans ma vie. Merci pour tous et que le bon Dieu, vous garde et vous protège.*

DJEBAILI YASMINE

# RESUME

*La tolérance aux pannes représente un challenge essentiel à relever et à tenir en compte dans la conception des protocoles de communication réseau ; surtout pour les applications temps réel où des décisions doivent être prises suite aux événements qui apparaissent et qui doivent être signalés impérativement par les réseaux de capteurs (WSNs : Wireless Sensor Networks).*

*Cette thèse s'oriente vers le contexte de la tolérance aux pannes. Elle nous a permis de proposer un nouveau protocole qui assure une continuité de fonctionnement même en cas de défaillance des nœuds ordinaires ou du CH (Cluster Head), ce dernier s'appelle FT-HEEP (Fault Tolerant Hybrid Energy Efficient Protocol), il est parmi les premiers protocoles qui traitent le problème de tolérance aux pannes en se basant sur l'approche Cross-layer. Notre protocole FT-HEEP est une extension du protocole HEEP, mais contrairement à HEEP qui ne propose aucun mécansime de tolérance aux pannes et qui n'opère qu'au niveau de la couche réseau, notre protocole est tolérant aux pannes et opère au niveau des trois premières couches du modèle OSI à savoir la couche physique, la couche MAC et la couche réseau. Au niveau de la couche physique, nous avons proposé un mécanisme qui permet de choisir les nœuds avec des liens de bonne qualité comme nœud de secours. Quant au niveau MAC, nous avons proposés un mécanisme permettant de choisir les liens les moins congestionnés. Et enfin nous avons proposé un mécanisme qui permet d'éviter les nœuds ou les CH's en panne et ce au niveau de la couche réseau.*

*Les évaluations par simulation de la solution proposée montrent un bon niveau de performances en termes de tolérance aux pannes.*

***MOTS-CLÉS :*** *RCSF, la tolérance aux pannes, estimation de la qualité des liens, détection de la congestion, RSSI.*

# ABSTRACT

*Fault tolerance represents an essential challenge in the design process of communication protocols, especially those related to real-time applications where appropriate decisions have to be applied, accordingly to the events that appear, and after being detected urgently by the sensor networks ( WSNs: Wireless Sensor Networks).*

*This thesis deals with the fault tolerance issue and proposes a routing protocol that ensures continuity of operation even in case of an ordinary node failure or a cluster head (CH) failure , the latter is called FT-HEEP and is among the few protocols that treat the fault tolerance problem based on the Cross-layer approach. Our FT-HEEP protocol is an extension of the HEEP protocol, but contrary to HEEP which offers no fault tolerance mechanism and which operates only at the network layer, our protocol is fault-tolerant and operates at the three first layers of the OSI model (the physical layer, the MAC layer and the network layer). At the physical layer, we have proposed a mechanism to choose good quality links as a backup node. As for the MAC layer, we have proposed a mechanism to choose the least congested links. And finally we proposed a mechanism that avoids the failed nodes or CHs and this at the network layer.*

*The simulation evaluations of the proposed solution show a good level of performance in terms of fault tolerance.*

***KEYWORDS**: Wireless sensor network, Fault tolerance, link quality estimation, Congestion Detection, RSSI*

# ملخص


*إن التسامح مع العطب يمثل تحديا أساسيا ينبغي أخذه بعين الاعتبار عند تصميم البروتوكولات ولا سيما بالنسبة لبعض التطبيقات التي يجب فيها إتخاذ القررات عند وقوع الأحداث والتي يجب الإشارة إليها من طرف شبكات الاستشعار ((ش.إ.ل) : شبكات الاستشعار اللاسلكية).*

*هذه الأطروحة تدخل في هذا الإطار، و تسمح باقتراح بروتوكول جديد يضمن استمرارية العمل حتى في حالة عطب رأس العقد أو العقد العادية ,هذا الأخير يسمى* FT-HEEP *وهو من بين البروتوكولات الأولى التي تعالج مشكلة التسامح مع العطب على أساس النهج المتعدد الطبقات . البروتوكول* FT-HEEP *هو امتداد لبروتوكول* HEEP*، ولكن على عكس* HEEP *الذي لايقدم أي آلية للتسامح مع العطب ويعمل فقط في طبقة الشبكة .*FT-HEEP *يقدم آلية للتسامح مع العطب ويعمل في الثلاث طبقات الأولى للموديل* OSI *(الطبقة الفيزيائية، الطبقة* MAC *و طبقة الشبكة). على مستوى الطبقة الفيزيائية، اقترحنا آلية لاختيار روابط ذات نوعية جيدة كعقدة احتياطية. أما فيما يتعلق بالمستوى* MAC*، فقد اقترحنا آلية لاختيار أقل الروابط ازدحاما. وأخيرا اقترحنا آلية تسمح بتجنب العقد أو رؤوس العقد المعطلة وهذا على مستوى طبقة الشبكة.*

*تقييم الحلول المقترحة عن طريق المحاكاة يظهر مستوى جيد من حيث التسامح مع العطب.*



***الكلمات المفتاحية*** *: شبكات الاستشعار اللاسلكية، التسامح مع العطب، تقدير جودة الروابط، كشف الإزدحام،* RSSI

# *CONTRIBUTIONS SCIENTIFIQUES*

**DJEBAILI Yasmine, BILAMI Azeddine,** A Cross-Layer Fault Tolerant Protocol with Recovery Mechanism for Clustered Sensor Networks, International Journal of Distributed System and Technologies (IJDST), 9(1), (2018). DOI: 10.4018/IJDST.2018010104.

**DJEBAILI Yasmine, BOURMADA Amal and BILAMI Azeddine.** A Hierarchical Fault Tolerant Routing Protocole for WSNs. International Conference on Networking and Advances Systems, Annaba (ICNAS'15), 2015.

**DJEBAILI Yasmine, BILAMI Azeddine & BOURMADA Amal :** Protocole de Routage pour l'Economie d'Energie et la Tolérance aux Pannes dans les RCSF, 1st JDTIC, Université de Batna, 4 et 5 Juin 2014.

**DJEBAILI Yasmine, BILAMI Azeddine :** Routing protocole for energy efficiency and fault tolerance in Wireless sensor network, JDI'2014, Université de Guélma, 3 et 4 Décembre 2014.

**BOURMADA Amal, BILAMI Azeddine & DJEBAILI Yasmine :** Protocole avec Différentiation de Service pour une QdS dans les RCSF, 1st JDTIC, Université de Batna, 4 et 5 Juin 2014.

**BOURMADA Amal, DJEBAILI Yasmine & BILAMI Azeddine :** An Energy aware Routing Protocole with Differentiated Services for WSN , CISC'14, Université de Jijel 9 et 10 Décembre 2014.

**DJEBAILI Yasmine & BOUAM Souheila :** Etude des performances de la méthode de localisation DV-HOP, JEESI'12, ESI, 16 Avril 2012.

# Table des Matières

# LISTE DES FIGURES

# LISTE DES TABLEAUX

# LISTE DES ABREVIATIONS ET SIGLES

LA signification des acronymes utilisés dans cette thèse est, en règle générale, précisée lors de leur première utilisation. Ci-après nous donnons tous ces acronymes, leur signification en anglais et (ou) une équivalence en français lorsque nécessaire.

| | |
|---|---|
| **RCSF** | Réseaux de capteurs sans fil |
| **WSN** | Wireless sensor Network |
| **LDA** | Location Dependent Address |
| **EUI** | End-system Unique Identifier |
| **LEACH** | Low-Energy Adaptive Clustering Hierarchy |
| **HEEP** | Hybrid Energy Efficient Protocol |
| **TCP** | Transmission Control Protocol |
| **UDP** | User Datagram Protocol |
| **ADCs** | Analog-to-Digital Convertors |
| **CSMA/CA** | Carrier Sense Multiple Access with Collision Avoidance |
| **RSSI** | Received Signal Strength Indication |
| **MAC** | Medium Access Control |
| **NS2** | Network Simulator version 2 |
| **QOS** | Quality Of service |
| **QDS** | Qualité De Service |
| **TDMA** | Time Division Multiple Access |
| **CH** | Cluster Head |
| **SB** | Station de base |
| **TTL** | Time To Life |
| **ACK** | Acknowledgement |
| **IP** | Internet Protocol |
| **GPS** | Global Positioning System |
| **PEGASIS** | Power-Efficient Gathering in Sensor Information Systems |
| **RTS** | Request To Send |
| **CTS** | Clear To Send |
| **RREP** | Route REPonse |
| **RREQ** | Route REQuest |
| **FIFO** | First In First Out |
| **FT-HEEP** | Fault Tolerant Hybrid Energy Efficient Protocol |

# INTRODUCTION GENERALE

*Les pannes sont la règle et non l'exception dans les réseaux de capteurs sans fil. Un nœud capteur est fragile et peut tomber en panne en raison de l'épuisement de la batterie ou de la détérioration par un événement externe. En outre, le nœud peut capter et transmettre des valeurs incorrectes en raison de l'influence de l'environnement sur son fonctionnement. Les liens sont également vulnérables et leur panne peut provoquer un partitionnement du réseau et un changement dans la topologie du réseau, ce qui conduit à une perte ou à un retard dans les transmissions des données. Dans le cas où les nœuds sont portés par des objets mobiles, ils peuvent être mis hors de portée de la communication. Les réseaux de capteurs sont également sujets à la congestion qui peut provoquer une perte des données. En plus de ces défaillances, les réseaux de capteurs présentent aussi des défaillances silencieuses qui ne sont pas connues à l'avance, et qui sont très liées au système. En revanche, les applications des RCSF, en particulier les applications temps réel, nécessitent un fonctionnement continu et fiable du système. Cependant, la garantie d'un fonctionnement correct d'un système pendant l'exécution est une tâche difficile. Cela est dû aux nombreux types de pannes que l'on peut rencontrer dans un tel système vulnérable et non fiable. Une approche holistique de la gestion des fautes qui aborde tous les types de fautes n'existe pas. Dans cette thèse, nous proposons un nouveau protocole de routage tolérant aux pannes combinant les trois notions présentées précédemment, à savoir la tolérance aux pannes, la qualité des liens et la congestion; cette combinaison est réalisée grâce à l'approche cross-layer (inter couches) qui consiste à concevoir des protocoles faisant interagir plusieurs couches de la pile protocolaire. En se basant sur ce principe, on peut développer des protocoles cross-layer qui traitent le problème de tolérance aux pannes au niveau de différentes couches, tout en respectant les caractéristiques des RCSF.*

*La suite de cette thèse est structurée autour de cinq (05) chapitres. Le premier chapitre présente les notions générales sur les réseaux de capteurs sans fil en commençant par la définition et l'anatomie d'un capteur, la définition d'un RCSF ainsi que les différentes topologies possibles de ces derniers. Ce chapitre permet également de faire une présentation des applications avant de passer en revue les défis et les problématiques de recherche qui font que cette technologie est d'actualité aussi bien dans le milieu universitaire qu'industriel. Il est terminé par une présentation des différents protocoles de routage conçus pour les RCSF.*

*Ensuite nous abordons le second chapitre où on détaillera le problème de tolérance aux pannes. Nous présentons une classification des différents types de pannes ainsi qu'une définition de la tolérance aux pannes. Ce chapitre nous permet aussi de discuter la procédure générale de tolérance aux pannes, ainsi que les différentes techniques de gestion de pannes.*

*Nous abordons dans le troisième chapitre un état de l'art sur les différents protocoles de routage tolérants aux pannes qui ont été proposés dans la littérature.*

*Dans le chapitre quatre, on présente un état de l'art sur la congestion, en insistant sur les différents protocoles de contrôle de la congestion proposés pour les RCSF.*

*Le chapitre 5 est réservé à notre contribution ; nous commençons par une présentation du concept d'architecture Cross-layer ainsi qu'une présentation du protocole HEEP sur lequel nous nous appuyons pour la proposition de notre nouveau protocole FT-HEEP. Ensuite, nous*

*présentons notre protocole avec tolérance aux pannes FT-HEEP, et on conclu ce chapitre avec une étude détaillée de ses performances par simulation et une conclusion.*

*En annexe, nous présentons l'environnement de simulation que nous avons utilisé pour évaluer les performances de notre contribution.*

CHAPITRE 1

# GENERALITES SUR LES RESEAUX DE CAPTEURS SANS FIL

## SOMMAIRE

# 1.1 Introduction

Au cours des dernières décennies, nous avons assisté à une miniaturisation du matériel informatique. Cette tendance à la miniaturisation a apporté une nouvelle génération de réseaux informatiques et télécoms présentant des défis importants et produisant en masse des systèmes d'une taille extrêmement réduite et embarquant des unités de calcul et de communication sans fil pour un coût réduit. Les réseaux de capteurs sans fil sont l'une des technologies visant à résoudre les problèmes de cette nouvelle ère de l'informatique embarquée et omniprésente, ils sont capables de générer et d'échanger des données d'une manière autonome et complètement transparente pour les utilisateurs.

Dans ce chapitre nous allons introduire et faire une description synthétique des réseaux de capteurs sans fil en présentant leurs évolutions, architectures, caractéristiques, leurs domaines d'applications variés, …etc.

# 1.2 Définition d'un capteur

Un capteur est un dispositif électronique ayant pour tâche de transformer une mesure physique environnementale observée en une mesure généralement électrique qui sera à son tour traduite en une donnée binaire exploitable et compréhensible par un système d'information et de la communiquer à un centre de contrôle via une station de base. Parmi les différents types de mesures enregistrées par les capteurs, on peut citer entre autres : la température, l'humidité, la luminosité, l'accélération, la distance, les mouvements, la position, la pression, la présence d'un gaz, la vision (capture d'image), le son, …etc.

Il comporte quatre unités de base représentées par une unité d'acquisition (dispositif de captage), une unité de traitement (un processeur), une unité de communication (un émetteur/récepteur radio) et une batterie. Le rôle de chacune des unités est défini dans les points suivants :

- **L'unité d'acquisition :** est généralement composée de deux sous-unités : les capteurs et Les convertisseurs analogique-numérique (ADCs : Analog-to-Digital Convertors). Les capteurs obtiennent des mesures numériques sur les paramètres environnementaux et les transforment en signaux analogiques. Les ADCs convertissent ces signaux analogiques en des signaux numériques.
- **L'unité de traitement :** comme le révèle son nom, cette unité est responsable de tous les traitements que doit effectuer un nœud capteur. Elle comprend deux interfaces : une interface avec l'unité d'acquisition et une autre avec le module de transmission. L'unité de traitement contrôle les procédures permettant au nœud capteur de réaliser les tâches d'acquisition et de stockage de données collectées, à travers un microcontrôleur (un simple processeur) et une mémoire limitée à quelques kilo octets.
- **L'unité de communication (Transceiver : transmitter-receiver) :** responsable de toutes les communications via un support de communication radio qui relie le nœud au réseau.
- **Batterie :** alimente les unités d'acquisition, de traitement et de communication. De plus, un nœud capteur peut être équipé d'autres composants supplémentaires tels qu'un système de localisation géographique GPS (Global Position System).

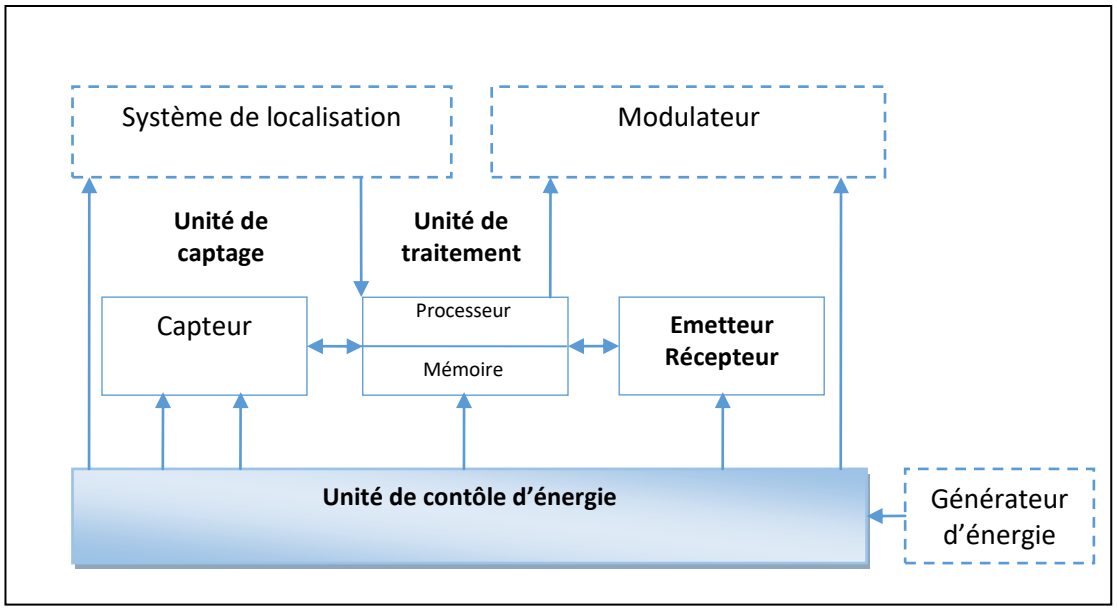



**Figure 1. 1** Anatomie d'un nœud capteur

## 1.3 Définition d'un réseau de capteurs

Un Réseau de Capteurs Sans Fil, Wireless Sensor Networks en anglais (WSN) est un système distribué de grande échelle mettant en communication un grand nombre de dispositifs très petits, autonomes, communément appelés "capteurs sans fil", ou simplement "capteurs". Dans ces réseaux, chaque nœud est capable de surveiller son environnement et de réagir, en cas de besoins, en envoyant l'information collectée, à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil.

Les capteurs sont des dispositifs de taille extrêmement réduite avec des ressources très limitées, autonomes, capables de traiter des informations et de les transmettre via les ondes radio (WiFi ou ZigBee par exemple), à une autre entité (capteurs, unité de traitements...) sur une distance limitée à quelques mètres. Les nœuds ont la capacité de jouer le rôle de routeurs. Un capteur analyse son environnement et propage les données récoltées aux capteurs appartenant à sa zone de couverture. Dans un scénario d'application classique, plusieurs nœuds capteurs sont déployés dans un certain environnement pour mesurer différents phénomènes physiques et faire remonter les informations collectées à une station de base distante, nommée aussi le nœud puits.

Dans le cas le plus simple, les capteurs communiquent directement avec la station de base. Cependant, dans le cas d'un réseau à grande échelle, les capteurs ne sont pas tous dans le voisinage du puits et les messages seront acheminés du nœud source vers le puits en transitant par plusieurs nœuds, selon un mode de communication multi-sauts comme l'illustre la Figure 1.2

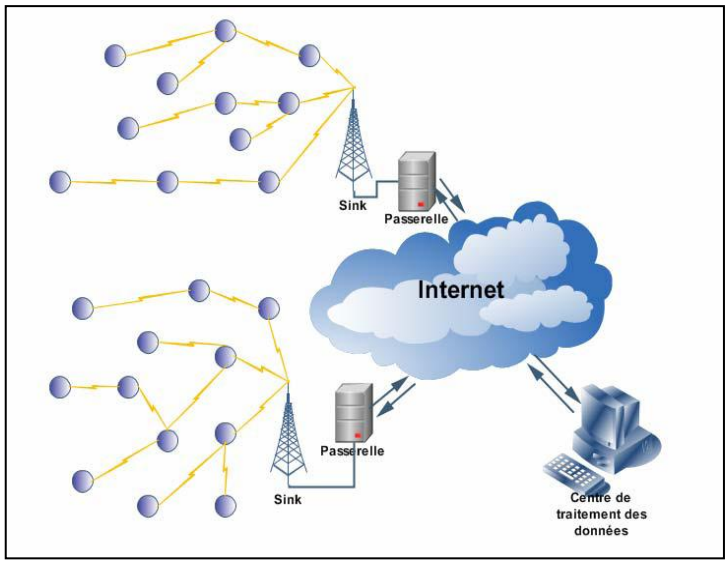



**Figure 1. 2** Schéma d'un réseau de capteurs sans fil

## 1.4 Les types des réseaux de capteurs sans fil

Selon des critères bien spécifiques, comme la mobilité, l'homogénéité des nœuds du réseau, la nature de l'application et le type des données captées, les RCSF peuvent être classés en plusieurs classes [6].

### 1.4.1 Selon le critère de mobilité

Les nœuds capteurs, ainsi que la station de base dans un réseau de capteurs sans fil peuvent être stationnaires ou bien mobiles. On parle alors des réseaux de capteurs statiques ou mobiles respectivement.

#### 1.4.1.1 Les réseaux de capteurs statiques

Dans les réseaux de capteurs statiques, et les nœuds capteurs et la station de base sont stationnaires ; ils gardent leurs positions initiales tout au long de leur durée de vie. Ce type de réseaux de capteurs est bénéfique dans certains types d'applications qui exigent que les capteurs soient placés dans des endroits stratégiques pour les contrôler. En effet, tel type de RCSF est caractérisé par une topologie statique, une localisation facile des nœuds dans le réseau et des techniques de routage assez simples.

#### 1.4.1.2 Les réseaux de capteurs mobiles

Contrairement aux RCSF statiques, dans les réseaux de capteurs mobiles, les capteurs et/ou la station de base ont la capacité de se mobiliser. La mobilité du capteur se produit soit quand le capteur est collé sur un objet mobile, soit quand le capteur s'auto-déplace (cas d'un capteur muni d'un moteur). La mobilité est indispensable dans les réseaux de capteurs destinés aux applications de suivi, par exemple,

quand les capteurs sont embarqués sur des véhicules, ou sur des animaux. Elle est (la mobilité) également avantageuse du point de vue coût d'investissement ; au lieu de déployer plusieurs nœuds statiques, un nombre minime de dispositifs mobiles est suffisant. Cependant, lorsque la mobilité est trop fréquente, elle ne peut être considérée comme un problème secondaire. Ainsi, le changement fréquent de la topologie complique les mécanismes de routage et de localisation.

## 1.4.2 Selon le critère d'homogénéité

Suivant ce critère, on observe deux types des réseaux de capteurs sans fil : les réseaux de capteurs homogènes et les réseaux de capteurs hétérogènes.

### 1.4.2.1 Les réseaux de capteurs homogènes

Un réseau de capteurs est dit homogène si tous les nœuds capteurs sont équivalents sur le plan capacités et contraintes (faibles ressources et durée de vie courte). C'est le type qu'on trouve souvent dans la majorité des applications des réseaux de capteurs, car ils répondent au besoin d'autonomie.

### 1.4.2.2 Les réseaux de capteurs hétérogènes

A l'encontre des réseaux de capteurs homogènes, les réseaux de capteurs hétérogènes comportent deux types de nœuds capteurs : les nœuds capteurs contraints (battery-powered) et les nœuds capteurs puissants non limités en ressources (particulièrement les ressources énergétiques comme ils sont directement liés à un secteur d'alimentation électrique). Dans ce type de RCSF, les nœuds contraints doivent préserver autant que possible leur réserve énergétique en minimisant les tâches les plus coûteuses en énergie tout comme la communication radio. Ainsi, les calculs et les traitements compliqués sont délégués aux nœuds puissants pour équilibrer la charge et maximiser la durée de vie du réseau. Bien que les RCSF hétérogènes soient plus avantageux que les RCSF ordinaires (homogènes), leur adoption est limitée à un nombre réduit d'applications. Cela est dû à la difficulté du déploiement des RCSF hétérogènes dans des milieux hostiles, isolés ou inaccessibles.

## 1.4.3 Selon le type de l'application

Le déclenchement du processus de captage de données dans un réseau de capteurs sans fil dépend des exigences applicatives et de l'importance de la donnée captée en elle-même. Donc, on distingue deux types de RCSF : temporels (time-driven) ou évènementiels (event-driven).

### 1.4.3.1 Les réseaux de capteurs temporels

Un réseau de capteurs temporel est approprié pour des applications qui nécessitent un prélèvement périodique des données. Tel est le cas par exemple dans les applications de monitoring (feu ou météo). Un écoulement en rafale, périodique, du trafic est très susceptible dans ce type d'applications. Par conséquent, des mécanismes de gestion raisonnable des ressources sont primordiaux.

### 1.4.3.2 Les réseaux de capteurs évènementiels

Dans certaines applications, les capteurs doivent réagir rapidement à des changements brusques des valeurs captées et donner des réponses immédiates à l'occurrence des évènements. Un prélèvement périodique des données est inadapté pour ce type de scénario.

## 1.4.4 Selon les données captées

Les données que récoltent les nœuds dans un réseau de capteurs peuvent être de type simple, comme ils peuvent être de type multimédia. De plus, un nœud capteur peut capter un seul type de données (exemple : que la température) ou plusieurs types à la fois (exemple : image, température et humidité).

### 1.4.4.1 Les réseaux de capteurs standards

Il s'agit des RCSF ordinaires où les données récoltées sont de types scalaires, comme par exemple : la température, l'humidité, la pression, etc. les RCSF de tel type partagent les caractéristiques déjà mentionnées.

### 1.4.4.2 Les réseaux de capteurs multimédia

Certaines applications des réseaux de capteurs, exigent que les données à capter soient de type multimédia (son, image ou vidéo) comme c'est le cas par exemple dans les applications médicales et les applications militaires. Néanmoins, les données multimédia sont reconnues pour être volumineuses et occupent donc, un espace mémoire important. Ainsi, des techniques de représentation différente que celles des données ordinaires sont nécessaires pour les données multimédia. Les réseaux de capteurs multimédia (ou Wireless Multimedia Sensor Networks: WMSN) requièrent des protocoles performants ainsi que des considérations particulières pour répondre à leurs défis en matière de qualité de service et de capacités de traitement exigées. D'autres spécificités liées aux WMSNs sont données ci-dessous :

- **le déploiement :** les nœuds dans les réseaux de capteurs standards sont souvent déployés aléatoirement. En revanche, dans les réseaux de capteurs multimédia, le déploiement des nœuds est généralement précis et étudié d'avance, particulièrement quand il s'agit du captage des images.
- **La puissance de traitement :** les traitements à effectuer sur les données scalaires sont faibles. Néanmoins, pour le cas des données multimédia, les nœuds capteurs effectuent des traitements intensifs ce qui demande plus de performance matérielle.
- **Qualité de service :** les réseaux de capteurs multimédia revendiquent suffisamment de bande passante ainsi qu'une faible latence pour qu'ils soient opérationnels, ce qui n'est pas le cas dans les réseaux de capteurs standards où la qualité de service est relâchée pour un besoin en un moindre coût et une faible dissipation des ressources.
- **Consommation d'énergie :** puisque la qualité de service et les traitements intensifs sont pratiquement gourmands en énergie, les mécanismes de gestion de la consommation énergétique dans les réseaux de capteurs multimédia doivent être très efficaces. A cet effet, on

note que dans ce cas, le remplacement des batteries des nœuds capteurs est souvent possible (tout dépend de la nature du champ de captage).

#### 1.4.4.3 Les réseaux de capteurs multimodaux

Un nœud capteur dans un RCSF multimodal peut récolter plusieurs informations de types différents où les types peuvent être scalaires ou multimédia. Par exemple, un nœud capteur peut capturer et la température et l'image. Ainsi, un seul nœud capteur multimodal peut remplacer tout un groupe de capteurs ordinaires. Ceci est particulièrement avantageux dans le cas où l'on veut avoir plus d'une information environnementale sur le même endroit d'intérêt.

## 1.5 Les topologies des réseaux de capteurs sans fil

Les topologies des réseaux de capteurs sont déterminées à partir des protocoles de routage utilisés pour l'acheminement des données entre les nœuds et le Sink [1]. Ces protocoles peuvent être hiérarchiques, plat (Flat) ou basé localisation [W1].

### 1.5.1. Topologie Hiérarchique

Les protocoles à topologie hiérarchique forment des réseaux dans lesquels un nœud central Sink (le niveau supérieur de la hiérarchie) est relié à un ou plusieurs autres nœuds qui appartiennent à un niveau plus bas dans la hiérarchie (deuxième niveau) avec une liaison point à point. Aussi, chacun des nœuds du deuxième niveau aura également un ou plusieurs autres nœuds de niveau plus bas dans la hiérarchie (troisième niveau) reliées à lui avec une liaison point à point. Chaque ensemble de nœuds forme une sorte de motif (Cluster). Le nœud central n'a aucun autre nœud au-dessus de lui dans la hiérarchie sauf le centre de traitement des données ou la passerelle si elle existe. Les nœuds du deuxième niveau jouent le rôle des passerelles entre ceux du troisième niveau et le Sink.

Dans ce cas, le routage devient plus simple, puisqu'il s'agit de passer par les passerelles pour atteindre le nœud destination.

Dans certains types de protocoles (tel que LEACH), un algorithme d'élection est exécuté dans chaque cluster, les nœuds élisent un d'eux pour être Clusterhead. L'élection est basée sur des critères tels que l'énergie disponible, la qualité de communication, et ainsi de suite, ou la combinaison de plusieurs d'entre elles. Le rôle du Clusterhead est la collecte des informations issues des nœuds et les renvoyer vers le Sink.

Un réseau basé sur une topologie hiérarchique doit avoir au moins trois niveaux dans sa hiérarchie, puisqu'un réseau avec un nœud central Sink et seulement un niveau hiérarchique au-dessous, forme une topologie en étoile.

Si les nœuds dans un réseau basé sur la topologie hiérarchique doivent effectuer un tel traitement sur les données transmises entre les nœuds dans le réseau, alors les nœuds qui sont à des niveaux plus élevés dans la hiérarchie doivent effectuer plus de traitement que les nœuds de niveau inférieur.

Dans le cas de LEACH, les informations sont transmises d'un nœud capteur vers le nœud Sink en passant par le Clusterhead déjà élu comme c'est illustré dans la figure 1.3.

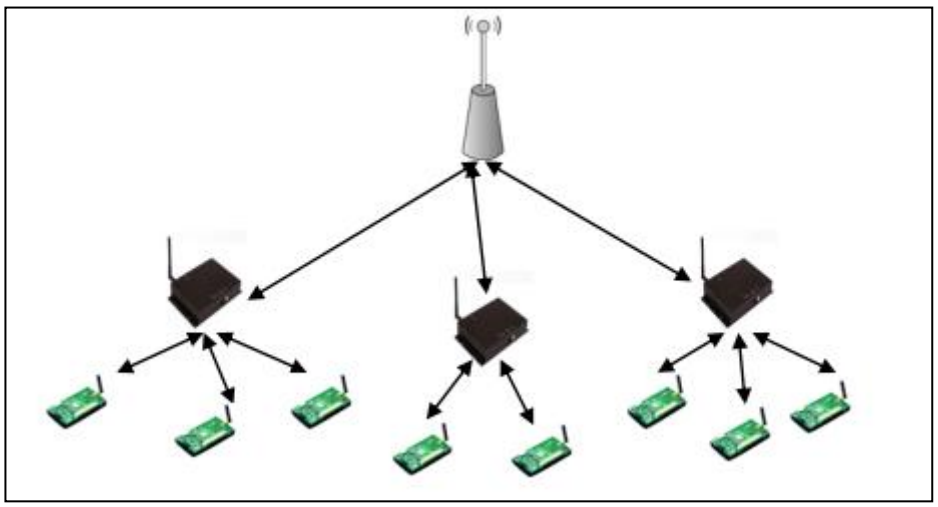


**Figure 1. 3:** Topologie Hiérarchique

À titre d'exemple des protocoles utilisant une topologie hiérarchique on peut citer le protocole LEACH (Low-energy Adaptive Clustering Hierarchy), CBRP (Cluster Based Routing Protocole), …etc

## 1.5.2. Topologie plate (Flat)

Les protocoles à topologie plate (flat) considèrent que tous les nœuds sont égaux, ont les mêmes fonctions, et peuvent communiquer entre eux sans devoir passer par un nœud particulier ou une passerelle. Seul un nœud particulier, le Sink, est chargé de la collecte des données issues des différents nœuds capteurs afin de les transmettre vers les centres de traitement.

En cas où la destination ne fait pas partie du voisinage de la source, les données seront transmises en utilisant les sauts multiples à travers les nœuds intermédiaires comme c'est illustré dans la figure 1.4. Ce type de réseau représente l'avantage de l'existence de différents chemins d'une source vers une destination et c'est pour remédier au problème de changement brusque de topologie ou la défaillance d'un nœud intermédiaire.

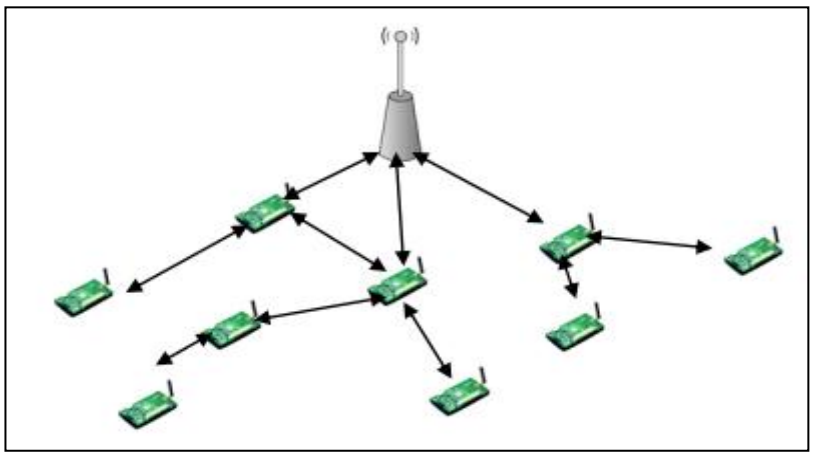


**Figure 1. 4 :** Topologie plate (Flat)

À titre d'exemple des protocoles utilisant une topologie plate on peut citer le protocole Direct Diffusion.

### 1.5.3 Topologie basée Localisation

Les protocoles à topologie basée localisation suppose que :

- Le réseau est partitionné en plusieurs zones de localisation.
- Chaque zone a son identifiant.
- Chaque nœud a un identifiant EUI (End-system Unique Identifier) et enregistre dynamiquement l'identifiant de la zone à laquelle il appartient temporairement.

L'information temporaire de localisation appelée LDA (Location Dependent Address) qui est un triplet de coordonnées géographiques (longitude, latitude, altitude) obtenues, par exemple, au moyen d'un GPS avec une précision dépendant du type de l'application. Une telle topologie exige l'implémentation d'un algorithme de gestion de localisation qui permet aux nœuds de déterminer les endroits approximatifs des autres nœuds. Ce type de topologie est mieux adapté aux réseaux avec une forte mobilité.

Avant d'envoyer ses données à un nœud destination, le nœud source utilise un mécanisme pour déterminer la localisation de la destination puis inclus l'identifiant de zone de localisation et du nœud destination dans l'entête du paquet à envoyer.

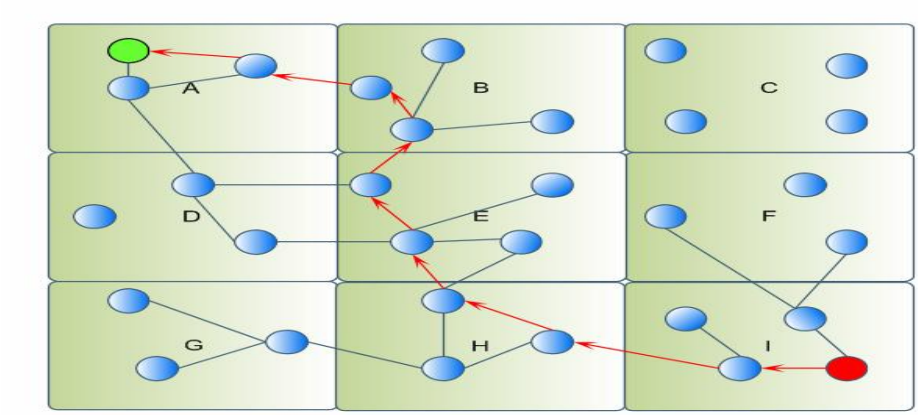



**Figure 1. 5 :** Topologie Basée Localisation

A titre d'exemple des protocoles utilisant une topologie basée localisation nous pouvons citer GEAR (Geographic and Energy Aware Routing) et LAR (Location-Aided Routing protocole).

## 1.6 Les domaines d'application des réseaux de capteurs sans fil

La miniaturisation, l'adaptabilité, le faible coût et la communication sans fil permettent aux réseaux de capteurs d'envahir plusieurs domaines d'applications. Ils permettent aussi d'étendre le domaine des

applications existantes. Parmi ces domaines où ces réseaux se révèlent très utiles et peuvent offrir de meilleures contributions, on peut noter le militaire, la santé, l'environnemental, et les maisons intelligentes...

### 1.6.1 Applications militaires

Les nœuds capteurs devraient fournir les services suivants :

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

Toutefois il faut noter que l'utilisation des RCSF à des fins militaires requiert une sécurité supérieure à tout autre domaine. En effet elle peut avoir une incidence voir mettre en jeu des vies humaines si des informations stratégiques sont récupérées par un ennemi [3].

### 1.6.2 Applications de santé

Certaines applications de santé des RCSF sont :

- Fourniture d'interfaces pour les handicapés.
- Repérage et surveillance des médecins et des patients dans les hôpitaux.
- Télésurveillance des données physiologiques humaines.

Cependant ces applications se heurtent à un problème de taille, à savoir la sécurité des informations transitant sur le réseau. D'un point de vue législatif, il est primordial que ces informations d'une part ne permettent pas d'authentifier le patient au regard du secret médical, d'autre part il faut pouvoir s'assurer que les données ne puissent être falsifiées. Ainsi une personne mal intentionnée pourraient envoyer de fausses informations sur une application sans protection, ce qui pourrait causer une mauvaise réaction du personnel médical et causer de graves troubles, voir la mort du patient [3].

### 1.6.3 Applications environnementales

Ces applications incluent :

- Le repérage des mouvements des oiseaux, des petits animaux et des insectes.
- La surveillance des conditions environnementales qui affectent les récoltes et le bétail.

- L'exploration planétaire.
- Alertes des catastrophes (incendie, séisme. . . ).
- La détection d'inondation.
- L'étude de pollution.

Le niveau de sécurité nécessaire dans des applications environnementales utilisant les RCSF peut paraître faible voir nul. Cependant ce serait oublier les risques de sabotage qui peuvent être le fait de personnes agissant dans un but gratuit (comme peut l'être la dégradation de bien publics) ou mercantile (les mesures environnementales peuvent impliquer des organisations et les mettre en porte-à-faux avec des conséquences économiques non négligeables) [3].

## 1.6.4 La domotique

Un autre type d'application dans lequel les réseaux de capteurs émergent, est la domotique. Dans cette application, le réseau de capteurs est déployé dans l'habitation. Le principe est que le réseau forme un environnement, dit pervasif où l'objectif étant de fournir toutes les informations nécessaires aux applications d'automatisation pour le confort, la sécurité et la maintenance dans la maison. Les capteurs sont incorporés dans différents dispositifs domestique (tel que : chauffage, système d'éclairage, dispositif d'incendie, alarme de détection d'intrusion, volet roulant, etc.), afin de répondre aux besoins de l'utilisateur, en lui permettant un contrôle plus aisé sur ces dispositifs localement ou à distance, par internet ou par satellite. Ce qui garantira aux habitants plus de confort, de sécurité, ainsi une facilité de maintenance, et d'un autre coté c'est l'un des moyens utilisés pour pouvoir minimiser les dépenses énergétique.

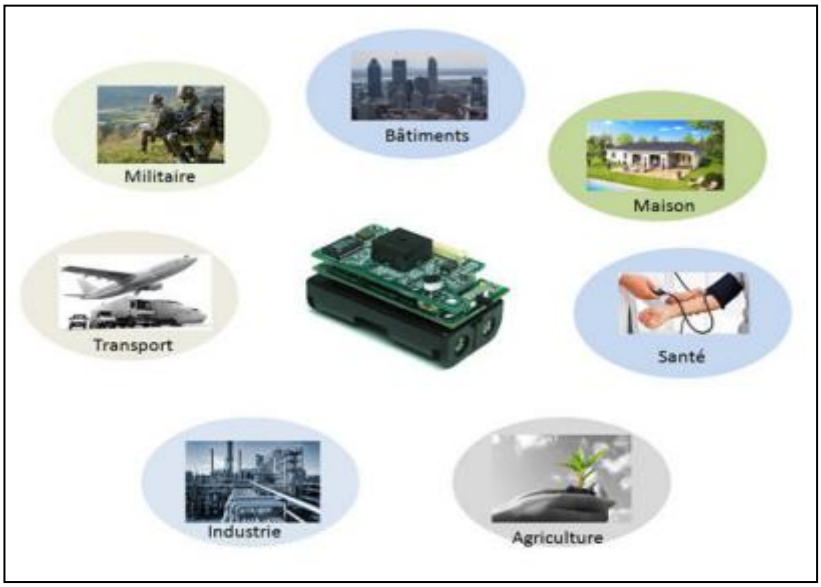



**Figure 1. 6 :** les domaines d'application des réseaux de capteurs sans fil.

## 1.7 Les facteurs de conception des réseaux de capteurs sans fil

La conception des réseaux de capteurs est influencée par de nombreux facteurs comme la tolérance aux pannes, les coûts de production, la consommation d'énergie, l'environnement ou la topologie du réseau. Ces facteurs représentent la base de la conception de protocoles ou d'algorithmes pour les réseaux de capteurs.

- **Tolérance aux pannes :** Les nœuds peuvent être sujets à des pannes dues à leur fabrication (ce sont des produits de série bon marché, il peut donc y avoir des capteurs défectueux) ou plus fréquemment à un manque d'énergie. Les interactions externes (chocs, interférences, …etc) peuvent aussi être la cause de leur dysfonctionnement. Afin que les pannes n'affectent pas la tâche première du réseau, Il faut mettre en place des mécanismes qui permettent d'augmenter le taux de disponibilité d'un réseau de capteurs [2].
- **Topologie du réseau :** En raison de leur forte densité dans la zone à observer, il faut que les nœuds capteurs soient capables d'adapter leur fonctionnement afin de maintenir la topologie souhaitée [2]. On distingue généralement trois phases dans la mise en place et l'évolution d'un réseau :
  - **Déploiement :** Les nœuds sont soit répartis de manière prédéfinie soit de manière aléatoire (largués en masse depuis un avion). Il faut alors que ceux-ci s'organisent de manière autonome.
  - **Post-Déploiement - Exploitation :** Durant la phase d'exploitation, la topologie du réseau peut être soumise à des changements dus à des modifications de la position des nœuds ou bien à des pannes.
  - **Redéploiement :** L'ajout de nouveaux capteurs dans un réseau existant implique aussi une remise à jour de la topologie.
- **La consommation d'énergie :** L'économie d'énergie est une des problématiques majeures dans les réseaux de capteurs. En effet, la recharge des sources d'énergie est souvent trop coûteuse et parfois impossible. Il faut donc que les capteurs économisent au maximum l'énergie afin de pouvoir fonctionner.

## 1.8 Les caractéristiques des réseaux de capteurs sans fil

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

- **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.

- **Topologie dynamique :** L'instabilité de la topologie des réseaux de capteurs est le résultat des trois facteurs essentiels suivants :
    - **La mobilité des nœuds :** les nœuds capteurs peuvent être attachés à des objets mobiles qui se déplacent librement et arbitrairement, introduisant ainsi une topologie instable du réseau.
    - **La défaillance des nœuds :** du fait de l'autonomie énergétique limitée des nœuds, la topologie du réseau n'est pas fixée (les nœuds qui épuisent leur énergie, sont considérés comme des nœuds inexistants).
    - **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.
- **Auto-organisation :** L'auto organisation s'avère très nécessaire pour ce type de réseau afin de garantir sa maintenance. Vu les différentes conséquences résultant de l'instabilité de la topologie du réseau de capteur, ce dernier devra être capable de s'auto-organiser pour continuer ses applications.
- **Scalabilité :** Les réseaux de capteurs peuvent contenir des centaines voire des milliers de nœuds capteurs. Un nombre aussi important engendre beaucoup de transmissions internodales et nécessite que le nœud «Sink» soit équipé d'une mémoire importante pour stocker les informations reçues.

## 1.9 Les différentes problématiques dans les réseaux de capteurs sans fil

Les recherches dans le domaine des réseaux de capteurs ont révélé plusieurs problématiques, parmi ces problématiques, nous citons [4] :

- **Routage :** Concevoir un protocole de routage performant en termes de minimisation de la consommation de l'énergie, du choix des routes optimales pour l'acheminement de l'information d'un capteur à la station de base et vice versa, de réduction du délai de délivrance des paquets...Ainsi le réseau doit passer à l'échelle sans que ses performances se dégradent.
- **Couche MAC :** La spécificité des réseaux de capteurs sans fil mobiles nécessite le développement de nouveaux protocoles MAC qui s'adaptent aux contraintes imposées par ces réseaux. Ceci est dans le but d'améliorer le débit, minimiser la consommation d'énergie, optimiser le partage du médium ainsi que minimiser le délai de délivrance des paquets.
- **Qualité de service :** Des protocoles au niveau de la couche MAC devraient être capables d'établir des priorités entre les flux, limiter les pertes de paquets vitaux pour la gestion du réseau, ou du moins en restreindre l'impact.
- **Cross-layer :** Dans les modèles classiques, les concepteurs essaient d'optimiser les performances au niveau d'une couche indépendamment des autres couches. Cependant, une telle indépendance est communément considérée comme trop onéreuse pour les réseaux de capteurs. Par conséquent, une coopération permettant un compromis entre

performances, dépendance et flexibilité doit être proposée pour optimiser les capacités du réseau en général.

- **Diffusion de l'information :** les protocoles de diffusion conçus pour les réseaux de capteurs doivent tenir compte de leurs spécificités ainsi que de leurs contraintes intrinsèques imposées. Ainsi, pour concevoir un protocole efficace, il faudrait assurer une couverture maximale des capteurs composant le réseau (taux d'accessibilité supérieur 90%), minimiser le nombre des réémetteurs et des réceptions redondantes ainsi que la consommation d'énergie.
- **Sécurité :** pour les applications qui exigent un niveau de sécurité assez élevé telles que les applications militaires, des mécanismes d'authentification, de confidentialité, et d'intégrité doivent être mis en place au sein de leur communauté. Les algorithmes de cryptographie conçus pour les réseaux de capteurs doivent tenir compte des ressources limitées que présentent ces réseaux.

## 1.10 La pile protocolaire des capteurs

Cette architecture est mise en place afin de structurer les protocoles de communication dans les RCSF. Ce modèle comprend 8 couches, cinq d'entre elles ont les mêmes tâches que celles du modèle OSI (Open System Interconnection), et trois autres couches pour assurer la gestion d'énergie (Power Management Plane), la gestion de la mobilité (Mobility Management Plane), et la gestion des tâches (Task Management Plane).

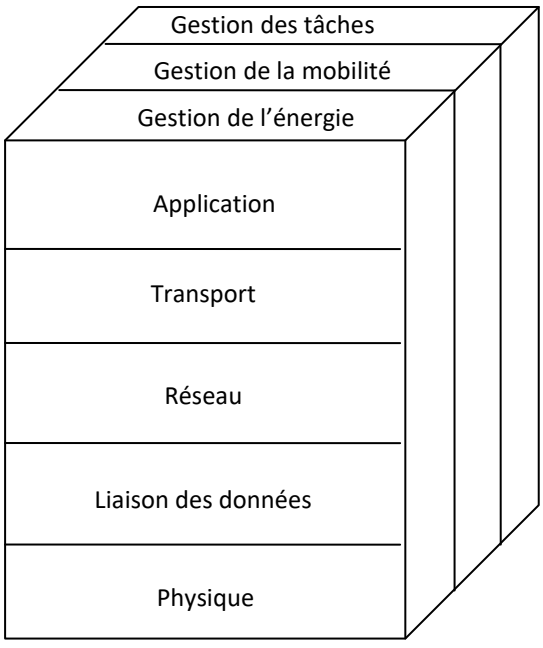



**Figure 1. 7** Modèle en couches pour la communication dans les RCSF.

### 1.10.1 Les rôles des couches

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

- **La couche physique :** Cette couche se charge de tout ce qui est spécifications des caractéristiques matérielles, la génération des ondes porteuses, la modulation de données et leur injection sur le support de transmission toute en sélectionnant les bonnes fréquences.
- **La couche liaison de données :** Spécifie comment les données sont expédiées entre deux nœuds dans une distance d'un saut. Elle est responsable de l'accès au media physique, du Couche Physique Couche Liaison de données Couche Réseau Couche Transport Couche Application Gestion de la mobilité Gestion de l'énergie Gestion des tâches, multiplexage des données, du contrôle d'erreurs. Elle assure la liaison point à point et multipoint dans un réseau de communication. Parmi les protocoles qui opèrent au niveau de cette couche on cite : SMAC (Self-organizing Medium Access Control for Sensor networks) et EAR (Eavesdrop And Register).
- **La couche réseau :** La couche réseau a pour but principal de baliser une route optimale en vue d'acheminer efficacement les données captées depuis leur source jusqu'au puits, tout en minimisant la dissipation énergétique des nœuds capteurs inclus dans le chemin. La tâche de routage au sein d'un réseau de capteurs est spécifique du fait que :
  - L'écoulement de données récoltées à partir de multiples sources vers une seule destination (la station de base).
  - La forte redondance de données et l'exigence de l'agrégation.
  - La nécessité d'une gestion soigneuse des ressources (énergie, mémoire, bande passante).
- **La couche transport :** Cette couche est chargée du transport de données, de la vérification de la qualité de la transmission et de la gestion des éventuelles erreurs. Dans le cas des réseaux de capteurs sans fil, la bonne qualité de transmission est souvent négligée car d'une part les pertes sont très probables avec un support de transmission sans fil et d'autre part, les mécanismes de gestion de la fiabilité sont trop lourd (tout comme le protocole TCP : Transmission Control Protocole). Ainsi, les pertes et les erreurs de transmission sont tolérables et peuvent même être camouflé par la redondance de données et l'agrégation. Le protocole UDP (User Datagram Protocole) qui fournit un service de transport en mode datagramme (sans connexion, sans gestion de congestion et sans fiabilité) est jugé d'être le protocole de transport le mieux adapté aux environnements capteurs en raison de sa faible empreinte mémoire et simplicité.
- **La couche application :** La couche application présente le niveau le plus proche des utilisateurs. de nombreux profiles d'applications peuvent être configurées et utilisées dans la couche application des réseaux de capteurs sans fil.

### 1.10.2 Les plans de gestion

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

- **Plan de gestion de mobilité :** Offre des mécanismes de détection et enregistrement des mouvements du nœud capteur. Ainsi, le nœud capteur peut garder trace de ses voisins.
- **Plan de gestion d'énergie :** Permet le contrôle de l'utilisation de la batterie, par exemple : après la réception d'un message, le capteur éteint son récepteur afin d'éviter la duplication du message déjà reçu. En outre, si le niveau d'énergie résiduelle devient bas, le nœud capteur diffuse à ses voisins une alerte les informant qu'il ne peut pas participer au routage et il préserve l'énergie restante pour le captage.
- **Plan de gestion des tâches :** Responsable de l'ordonnancement des tâches de captage de données dans une région spécifique. Il n'est pas nécessaire que tous les nœuds capteurs de cette région effectuent la tâche de captage au même temps ; certains nœuds capteurs la font plus que d'autres suivant que l'énergie résiduelle leur soit suffisante ou non.

## 1.11 Les protocoles de routage pour les réseaux de capteurs sans fil

### 1.11.1 Classification des protocoles de routage

#### 1.11.1.1 Les classes de protocoles de routage :

Cette section présente trois classes principales de protocoles de routage dédiés aux RCSF [78], à savoir les protocoles utilisant le routage plat, le routage hiérarchique ou le routage géographique.

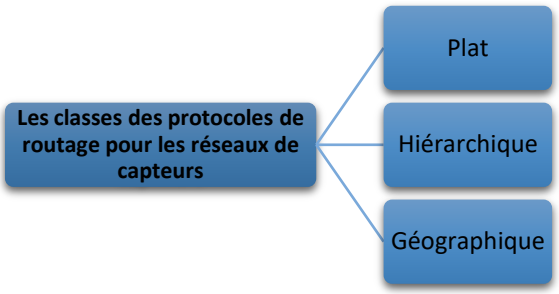



**Figure 1. 8:** Classification des protocoles de routage selon la structure du réseau

### *a. Les protocoles plats*

Ces protocoles supposent qu'il est difficile d'avoir des identifiants comme les adresses MAC ou IP pour pouvoir communiquer entre les nœuds capteurs. Ils ne demandent pas un mécanisme d'adressage. Dans ce type de protocoles les informations sont propagées de proche en proche. Ils Envoient une annonce des données avant d'envoyer les données elles-mêmes, les voisins intéressés demandent les données annoncées, les données sont ensuite envoyées.

- **SPIN (Sensor Protocoles for Information via Negotiation)**

Le protocole SPIN [5] permet de disséminer des informations sur le réseau de manière ciblée. Le fonctionnement du protocole SPIN permet de réduire la charge du réseau par rapport aux méthodes de diffusion traditionnelles telles que l'inondation ou l'algorithme de Gossiping .

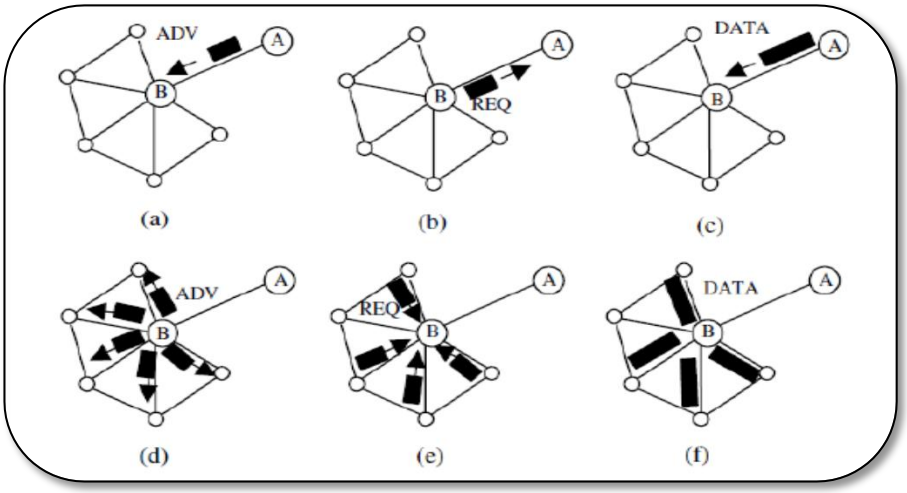



**Figure 1. 9 :** Fonctionnement du protocole SPIN

Le protocole SPIN utilise essentiellement trois types de paquets ADV/REQ/DATA. Un nœud voulant émettre une donnée commence par envoyer un paquet ADV. Ce paquet ADV consiste d'une méta-données sur les données à émettre. Les méta-données peuvent décrire plusieurs aspects comme le type des données et la localisation de son origine. Les nœuds qui reçoivent ce paquet vérifient si les données les intéressent. Si oui, ils répondent par un paquet REQ. Le nœud qui a initié la communication envoie alors un paquet DATA pour chaque réponse REQ reçue (voir la Figure 1.9). Un nœud peut parfaitement ne pas répondre aux messages ADV, par exemple dans le but d'économiser son énergie. Ensuite chaque nœud qui fait office de relais peut très bien agréger ses propres donnés aux données qui sont déjà contenues dans le paquet.

- **Direct diffusion**

Direct diffusion [6] est l'inverse de SPIN : les nœuds intéressés par une donnée diffusent une requête. Les nœuds voisins prennent en compte cette requête, répondent en fonction et rediffusent à leur tour la requête.

## *b. Protocoles hiérarchiques*

Construction de clusters (groupe de nœuds) avec un chef par cluster qui se chargera de transmettre les messages générés par son cluster aux autres chefs de clusters pour atteindre la destination finale. Le choix du chef de cluster (cluster head) est fait soit à tour de rôle, soit selon le nombre de voisins en considérant comme cluster head le nœud avec le plus de voisins, soit selon le niveau de l'énergie résiduelle...

- **LEACH (Low Energy Adaptive Clustering Hierarchy)**

LEACH [7] est l'un des protocoles hiérarchiques les plus populaires. Ce protocole suit le modèle « Time-Driven » et utilise un clustering distribué (la formation des clusters et l'élection des cluster-heads se font au niveau des nœuds). LEACH suppose que les nœuds sont homogènes et que l'acheminement des paquets vers la station de base se fait en un seul saut via les cluster-heads. Les nœuds ont la possibilité de devenir cluster-heads en se basant sur des probabilités d'élection.

LEACH s'exécute en cycles « rounds », comportant chacun quatre phases qui sont :

- **Phase des annonces :** C'est la phase où les clusters sont formés et les cluster-heads sont élus pour une période déterminée « round ». Cette élection se fait d'une manière cyclique dans le but d'équilibrer la dissipation d'énergie. Durant cette étape, chaque nœud N choisit aléatoirement un nombre dans l'intervalle [0,1], si ce nombre est inférieur à un seuil T(N) alors le nœud est élu cluster-head. T(N) est défini comme suit [7] :

  Où :

$$T(N) = \begin{cases} \dfrac{p}{1 - p[r \bmod (\frac{1}{p})]} & si\ N \in G \\ 0\ sinon \end{cases}$$

  r : numéro du round courant ;

  p : pourcentage des nœuds désirant devenir cluster-head ;

  G : ensemble de nœuds n'ayant pas été élus cluster-heads durant les 1/P dernières périodes.

  Dès que les cluster-heads sont élus, chacun d'eux envoie un message de notification aux autres nœuds du réseau. Ces nœuds décident donc de leur appartenance à un cluster selon l'amplitude du signal reçu en choisissant le signal le plus fort.

- **Phase de création des clusters :** Une fois la décision prise, chaque nœud doit informer son cluster-head de son choix par l'envoi d'un paquet d'affiliation.

- **Phase de création de l'ordonnancement :** Chaque cluster-head ayant reçu les messages des nœuds désirant appartenir à son cluster, diffuse un ordonnancement TDMA aux membres de son cluster en attribuant à chaque membre un intervalle de temps durant lequel il pourra communiquer ses données.

- **Phase de transmission de données : Après** l'établissement de l'ordonnancement, les membres communiquent et transmettent les données vers leurs cluster-heads (CHs) en un seul saut durant les slots qui leur ont été consacrés. Les CHs agrègent les données reçues et les transmettent vers la station de base.

- **Les variantes de LEACH**

Dans [12], les auteurs ont comparé les réseaux homogènes et hétérogènes en termes de dissipation d'énergie dans tout le réseau et ils ont analysé les performances des réseaux à un saut et ceux à sauts multiples. Ils ont choisi pour cela LEACH comme représentant des réseaux homogènes et ils l'ont comparé avec un réseau hétérogène à un saut. Les auteurs ont constaté que l'utilisation des communications à un saut entre les membres d'un cluster et leur cluster-head correspondant pourrait ne pas être le bon choix quand l'index $k$ de perte de propagation ($k > 2$) pour les communications intra-clusters est plus grand. D'autre part, LEACH pourrait produire des clusters possédant une taille importante dans les réseaux denses et des clusters dont la taille est limitée dans les réseaux de petites tailles. Dans ces deux cas, les cluster-heads pourraient rapidement épuiser leur puissance de batterie. Dans les réseaux denses, les cluster-heads coordonnent entre plusieurs membres des clusters alors que dans les réseaux de petites tailles, les cluster-heads sont placés loin de la station de base ce qui nécessite des transmissions de forte puissance.

Dans le même article, les auteurs ont proposé une version améliorée de LEACH appelée M-LEACH [8] (Multi-hop LEACH), dans laquelle les membres d'un cluster peuvent être à plus d'un saut de leur cluster-head correspondant et communiquent avec lui en mode multisaut. Ainsi, ils ont illustré les cas dans lesquels M-LEACH surpasse LEACH. Cependant, cette version proposée exige que chaque capteur doit être capable d'agréger les données, ce qui augmente l'overhead pour tous les capteurs.

Pour améliorer cette stratégie, dans [9], les auteurs se sont focalisés sur les réseaux de capteurs hétérogènes, dans lesquels deux types de capteurs sont déployés : capteurs de grandes capacités (Super Sensor) et capteurs simples. Les capteurs de grandes capacités ont des capacités de traitement et de communication si élevées et agissent comme cluster-heads, alors que les autres sont des capteurs simples avec une puissance limitée, affiliés au cluster-head le plus proche dans leur voisinage et communiquent avec lui directement ou en mode multi-saut.

En outre, une autre variante de LEACH appelée LEACH-C [7] a été conçue pour améliorer les performances de LEACH. Cette variante utilise une architecture centralisée pour choisir les cluster-heads tout en impliquant la station de base et l'information de localisation des capteurs. Cependant, elle augmente considérablement l'overhead du réseau puisque tous les capteurs devront envoyer leurs informations de localisation à la station de base en même temps pendant chaque phase d'élection de cluster-heads. Plusieurs travaux présentés dans la littérature ont prouvé qu'une telle architecture centralisée ne supporte pas le passage à l'échelle et est plus particulièrement appropriée à des réseaux de petite taille.

D'une manière similaire à LEACH-C, BCDCP [10] (Base-Station Controlled Dynamic Clustering Protocole) implique le niveau d'énergie des capteurs envoyé à la station de base pour construire des clusters homogènes durant la phase d'installation (1ière phase). La station de base choisit aléatoirement les cluster-heads tout en garantissant une distribution uniforme de leurs emplacements dans la zone d'intérêt dans laquelle ils sont déployés, et exécute un algorithme itératif de fusion pour trouver le nombre optimal de clusters. Puis, elle établit les routes inter-clusters (CH-to-CH) pour l'acheminement des données d'un cluster-head à un autre, et crée un schedule pour chaque cluster qui le diffuse dans le réseau. Durant la deuxième phase, les cluster-heads transmettent les données collectées à la station de base par des chemins CH-to-CH [11]. Néanmoins, BCDCP présente les mêmes limitations que LEACH-C puisqu'il utilise une architecture centralisée pour élire les cluster-heads.

Les auteurs dans [12] ont développés une autre variante de LEACH appelé LEACH-F (LEACH with fixed clusters) dans laquelle la formation des clusters se fait une seule fois et le cluster-head change de position à chaque fois, l'avantage d'un tel développement est qu'il n'y a pas de set-up phase à chaque fois. LEACH-F utilise le même algorithme d'élection que LEACH-C. Son inconvénient est qu'il n'ajoute pas les nouveaux nœuds au cluster et ne prend pas en charge la mobilité des nœuds.

Les techniques de clustering que nous avons présentées dans cette section, quoi qu'elles préconisent une solution garantissant l'équilibre des charges dans l'élection des cluster-heads, ont un impact négatif sur les cluster-heads, puisque leur choix se fait aléatoirement. Or, ces derniers consomment leur énergie plus rapidement qu'un nœud ordinaire puisqu'ils supportent des fonctions additionnelles comme l'agrégation des données et le routage. Le choix d'un cluster-head qui a un niveau d'énergie plus faible, pourrait vite devenir un goulet d'étranglement de son cluster. D'autre part, dans la phase de reconstruction des clusters, un overhead de communications et de calculs est généré puisque tous les capteurs envoient simultanément leurs niveaux d'énergie à la station de base et la connaissance appropriée de la topologie du réseau est exigée.

- **PEGASIS (Power-Efficient Gathering in Sensor Information Systems)**

Dans [13], Lindsey et Raghavendra ont proposé une version améliorée de LEACH appelée PEGASIS. L'idée principale de PEGASIS est de former une chaîne entre les nœuds de sorte que chaque nœud

reçoit et communique à un voisin proche. Les données collectées sont transmises d'un nœud à un autre qui les agrège jusqu'à ce qu'elles arrivent à un nœud particulier qui les transmet à la station de base. Les nœuds qui transmettent les données à la station de base, sont choisis tour à tour selon un round-robin dans le but est de réduire l'énergie moyenne dépensée par un nœud durant une période (round). Contrairement à LEACH, PEGASIS évite la formation des clusters et procure à un seul nœud dans la chaîne l'envoi de données à la station de base.

Les résultats de simulation ont montré que PEGASIS peut prolonger de deux à trois fois la durée de vie d'un réseau de capteurs relativement à LEACH en fonction du critère choisi pour évaluer la durée de vie d'un réseau i.e. quand 1%, 20%, 50% ou 100% des nœuds épuisent leurs batteries. Un tel gain de performance est réalisé par l'élimination de l'overhead causé par le processus de formation de clusters dans LEACH, et par réduction du nombre de transmissions et de réceptions par utilisation de l'agrégation de données. Bien que l'overhead du clustering soit évité, PEGASIS exige toujours un ajustement dynamique de la topologie puisqu'un nœud devrait connaître le niveau d'énergie de ses voisins avant de relayer ses données. Cependant, un tel ajustement de la topologie pourrait causer un overhead important en particulier dans les réseaux les plus utilisés. En outre, PEGASIS suppose que tout nœud est capable de communiquer directement avec la station de base. Or, cette supposition est loin de la réalité car les capteurs communiquent généralement en mode multi-sauts pour atteindre la station de base. D'autre part, PEGASIS suppose que tous les nœuds maintiennent une table contenant les localisations de tous les autres nœuds dans le réseau. En résumé, PEGASIS est adapté seulement aux capteurs sans fil dont les nœuds sont immobiles. Son évaluation dans des environnements mobiles pourrait dégrader considérablement ses performances.

### *c. Protocoles basés sur la position*

Dans les réseaux de capteurs, on considère que la position du nœud est plus importante que son identité (adresse). Ce type de protocoles considère que les nœuds connaissent leur position respective et sont capables de connaître la position des autres nœuds. Ainsi, cette information est utilisée pour diriger les messages vers la région dans laquelle se trouve la destination.

- **GEAR (Geographic and Energy Aware Routing)**

Ce protocole de routage découpe le réseau en régions. Chaque nœud connait le coût pour atteindre chaque région. L'acheminement des paquets suit les étapes suivantes :

- acheminer le paquet jusqu'à la région, en envoyant le paquet au nœud le plus proche de la région parmi ses voisins et ayant le niveau d'énergie résiduelle le plus élevé (fonction de distance et d'énergie).

- acheminer le paquet dans la région de destination par une sorte de diffusion si le nombre de nœud n'est pas élevé, sinon la région est découpée en sous-région et le paquet est transmis individuellement à chaque sous-région.

Chaque paquet contient la région destination. Chaque nœud connait sa position, son énergie résiduelle, la position et l'énergie résiduelle de ses voisins (à la demande). Un lien existe entre 2 nœuds quand ils sont à portée et leur niveau d'énergie leur permet d'effectuer l'envoi.

### *d. Protocoles basés sur la QoS*

Le principe des protocoles de routage avec QoS se base sur le fait que le réseau doit être capable de satisfaire certaines métriques (latence, énergie des nœuds, bande passante, et.) tout en acheminant le maximum de données vers la station de base. Il existe plusieurs protocoles de ce type dans la littérature

- **SAR (Sequential Assignment Routing)**

SAR est l'un des tous premiers protocoles ayant introduit la QoS dans ses décisions de routage. Celle-ci dépend de trois facteurs : ressources énergétiques, QoS sur chaque chemin, et niveau de priorité de chaque paquet. Pour éviter l'échec sur un simple chemin, une approche de routage par chemins multiples et des méthodes de restauration locale du chemin sont employées par SAR. En tant que tel, SAR est un protocole de routage qui vise à réaliser l'efficacité énergétique et la tolérance aux fautes. De manière synthétique, SAR calcule une métrique pondérée de la QoS comme résultat du cumul de métriques QoS et d'un coefficient de poids associé au niveau de priorité du paquet à router. L'objectif de SAR est de minimiser la moyenne de la métrique pondérée de la QoS durant toute la durée de vie du réseau. Bien que ceci assure une tolérance aux fautes et un recouvrement facile, SAR souffre de la surcharge liée à la maintenance des tables et des états de chaque nœud capteur, particulièrement lorsque le nombre de nœuds capteurs devient très grand.

- **SPEED**

SPEED est un autre protocole de routage avec QoS pour RCSF qui fournit des garanties temps-réel de bout-en-bout [14]. Chaque nœud du réseau maintient l'information sur ses voisins et emploie l'acheminement géographique pour trouver les chemins reliant les sources aux destinations. En outre, SPEED tente d'assurer une certaine vitesse de relais des nœuds voisins, de sorte que le délai de bout-en bout du paquet soit proportionnel à la distance entre la source et la destination.

SPEED fournit aussi le moyen de faire éviter aux paquets les zones congestionnées du réseau en se basant sur des messages de contre-pression. Ces messages sont envoyés par les nœuds congestionnés à leurs nœuds ascendants pour le choix de chemins alternatifs. Comparé à d'autres protocoles, SPEED assure un meilleur fonctionnement en termes de délai moyen de bout-en-bout, de surcharge de contrôle et du taux de réussite dans l'acheminement des paquets. Cependant, SPEED ne considère aucune autre métrique d'énergie dans ses décisions de routage. De plus, il traite un vide comme étant une congestion passagère dans le réseau. Ce traitement n'est pas réaliste, un vide peut en effet rester

longtemps dans le réseau. Il ne peut disparaitre qu'après un redéploiement de nouveaux nœuds capteurs ou suite à un mouvement de quelques nœuds voisins mobiles pour compléter la zone vide.

### 1.11.1.2 Les sous-classes des protocoles de routage

Chaque classe principale de protocoles (Figure 2.8) peut à son tour donner naissance à cinq sous-classes, selon la stratégie de routage du protocole : routage basé sur les chemins multiples, routage basé sur les requêtes, routage basé sur la négociation entre les nœuds, routage basé sur la cohérence des données, ou routage avec qualité de service. Ces sous-classes, avec des exemples de protocoles existants, sont données dans la figure 2.10.

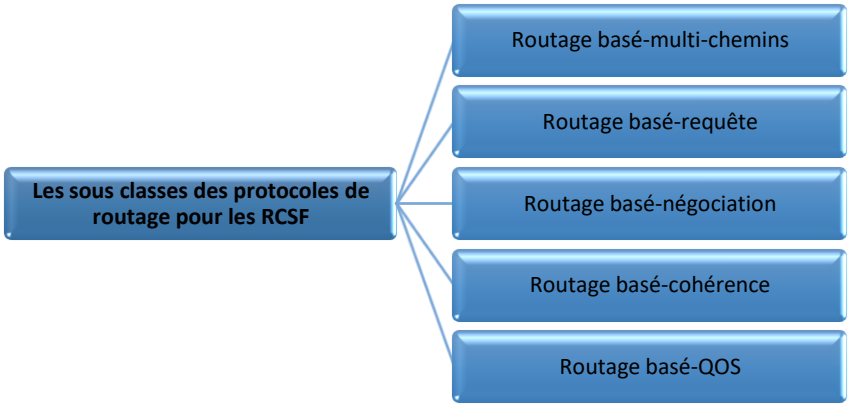



**Figure 1. 10 :** Classification suivant la stratégie de routage du protocole.

**a. Routage basé sur les chemins multiples**

Les protocoles de cette sous-classe utilisent des chemins de routage multiples au lieu d'un chemin unique entre une source et une destination. La tolérance aux fautes d'un protocole est mesurée par la vraisemblance qu'un chemin alternatif existe entre une source et une destination lorsque le chemin principal échoue. Cette tolérance peut être renforcée en découvrant des chemins multiples entre la source et la destination aux dépens d'une consommation énergétique et d'un trafic de contrôle supplémentaires. Ces chemins alternatifs sont maintenus en veille par la source en envoyant des messages périodiques. Par conséquent, la fiabilité du réseau peut être augmentée tout en accusant une surcharge de contrôle supplémentaire pour garantir la validité des chemins alternatifs.

**b. Routage Basé sur les requêtes**

Dans ce type de protocole, le puits propage des requêtes vers les nœuds capteurs. Ces derniers ayant des données à transmettre, répondent en émettant les données via le chemin inverse des requêtes. Les deux protocoles : Directed Diffusion [6] et Rumour Routing [15] se basent sur ce principe.

**c. Routage basé sur la négociation entre les nœuds**

Ces protocoles emploient des descripteurs de données à un niveau élevé afin d'éliminer la transmission des données redondantes sur la base de la négociation. Des décisions de communication sont également prises sur la base des ressources disponibles au niveau des nœuds capteurs. SPIN [5] sont des exemples de protocoles de routage via la négociation. La motivation principale réside dans le fait que l'utilisation de l'inondation pour dissémination produira la duplication des données envoyées, ainsi les nœuds recevront les copies doubles des mêmes données. Cette opération consomme un surplus d'énergie et de traitement en envoyant les mêmes données par différents capteurs. SPIN sont conçus pour disséminer les données d'un nœud à tous les autres nœuds, en supposant que ces capteurs sont de potentielles stations de base. Donc, l'idée principale du routage via la négociation est de supprimer l'information double et d'empêcher l'envoi des données redondantes au prochain capteur ou à la station de base, en échangeant une série de messages de négociation avant même la transmission effective des données.

**d. Routage basé sur la 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 (délai de transmission ou niveau de fiabilité) et doit faire l'équilibre avec la consommation d'énergie.

Le protocole SPEED [14] est l'un des premiers protocoles géographiques basé sur la qualité de service.

**e. Routage basé sur la cohérence des données**

Le traitement de la cohérence des données est une phase importante dans le fonctionnement des RCSF. Par conséquent, des algorithmes de routage utilisent différentes techniques pour traiter la cohérence (ou la non-cohérence) des données circulant dans le réseau. En général, les nœuds capteurs coopèrent entre eux afin de réaliser ce traitement. Dans le routage basé sur la non-cohérence des données, ces dernières sont envoyées aux nœuds agrégateurs du réseau après avoir reçu le traitement minimum qui inclut la suppression des doublures. Pour exécuter un routage efficace en énergie, le traitement de la cohérence des données est normalement choisi par le concepteur du protocole.

## 1.12 Conclusion

Dans ce chapitre nous avons introduit les réseaux de capteurs sans fil, ou nous avons présenté tous les aspects liés à cette technologie et comment elle est utilisée dans tous les domaines d'application. Outre ce type de RCSF classiques que nous avons présenté et qui étaient déployés pour des applications privées où les données de captage étaient récupérables à partir des stations de base, la nouvelle génération des RCSF est désormais invitée à intégrer l'Internet. Dans ce cas, les rapports des capteurs intégrés à Internet sont accessibles de n'importe où et n'importe quand, à partir d'un autre bout connecté également à Internet. Donc, l'accès et la récupération des données de captage deviennent ubiquitaires.

Les RCSF jouent un rôle très intéressant dans l'Internet des objets. En effet, les capteurs permettent la représentation des caractéristiques dynamiques (température, humidité, pression, mouvements, …)

des objets et des endroits du monde réel dans le monde virtuel représenté par le réseau Internet global. Ainsi, avec l'incorporation des réseaux de capteurs dans l'Internet, Les capteurs deviennent des serveurs (fournisseurs de services) dans ce que l'on désigne par le web des objets (dit WoT pour Web of Things) [73]. Ainsi, les services (applications) des RCSF se rajoutent à l'ensemble des services et applications de l'Internet de futur qui réunira une variété de réseaux fortement hétérogènes (que ça soit sur le plan matériel ou logiciel), soumis à des contraintes différentes et qui sont déployés pour diverses applications, afin d'en avoir un monde réel très sophistiqué.

CHAPITRE 2

# LA TOLERANCE AUX PANNES DANS LES RESEAUX DE CAPTEURS SANS FIL

## SOMMAIRE

# 2.1 Introduction

La limitation d'énergie dans les capteurs sans fil, et les environnements hostiles dans lesquels ils pourraient être déployés, sont des facteurs qui rendent ce type de réseaux très vulnérables. Ainsi la perte de connexion sans fil peut être due à une extinction d'un capteur suite à un épuisement de sa batterie, ou tout simplement à une destruction physique accidentelle ou intentionnelle par un ennemi.

Par ailleurs, l'absence de sécurité physique pour ce type de capteurs, et la nature vulnérable des communications radios sont des caractéristiques qui augmentent les risques de pannes sur ce type de réseau. Etant donné que les réseaux de capteurs reposent sur des protocoles de communication ad hoc, il est donc nécessaire de considérer la tolérance aux pannes comme critère indispensable dans la conception de ces protocoles.

Ce chapitre s'articulera sur la notion de tolérance aux pannes dans les réseaux de capteurs où nous commencerons par sa définition. Après une classification des pannes dans ce type de réseaux. Par la suite, nous présentons les diffèrentes techniques de gestion des pannes dans les RCSF.

# 2.2 Classification des pannes

Il est utile de classifier les pannes selon différents critères. Le schéma suivant montre une classification générale selon la durée, la cause ou le comportement d'une panne :

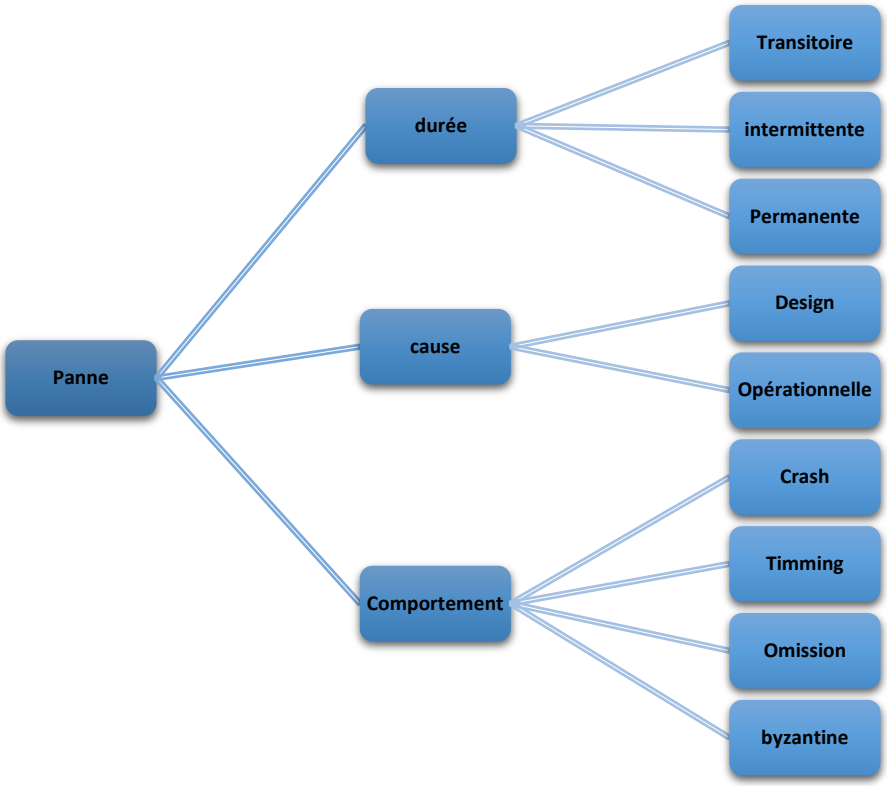



**Figure 2. 1 :** Classification des pannes.

## 2.2.1 Pannes selon la durée

Basée sur sa durée, la panne peut être classifiée en :

- **Transitoire** : conséquence d'un impact environnemental temporaire, elle peut éventuellement disparaître sans aucune intervention.
- **Intermittente** : variante de la panne transitoire, elle se produit occasionnellement et de façon imprévisible. Elle est généralement due à l'instabilité de certaines caractéristiques matérielles ou à l'exécution du programme dans un espace particulier de l'environnement.
- **Permanente** : continue et stable dans le temps, la panne permanente persiste tant qu'il n'y a pas d'intervention externe pour l'éliminer. Un changement physique dans un composant provoque une panne matérielle permanente.

## 2.2.2 Pannes selon la cause

On distingue deux types de pannes selon leur cause :

- **Panne de design :** due à une mauvaise structuration du réseau ou du composant en particulier. En pratique, ce genre de panne ne devrait pas exister grâce aux tests et simulations avant la réalisation finale du réseau.

- **Panne opérationnelle :** qui se produit durant le fonctionnement du système. Elle est généralement due aux causes physiques. En outre, on peut distinguer, spécialement pour les réseaux de capteurs, trois principales causes :

  - **Energie :** l'épuisement de la batterie cause l'arrêt du capteur. La consommation d'énergie est très importante pour déterminer la durée de vie d'un nœud capteur, et donc de tout le réseau ;
  - **Sécurité :** la destruction physique accidentelle ou intentionnelle par un ennemi peut être une cause de panne. L'absence de sécurité dans les réseaux de capteurs augmente le risque des pannes de ce type ;
  - **Transmission :** la nature vulnérable de transmission radio, la présence d'obstacles dans les environnements hostiles ainsi que les interférences électriques peuvent être la source d'une faute lors du transfert de données.

## 2.2.3 Pannes selon le comportement résultant

Après l'occurrence d'une panne, on distingue quatre différents comportements possibles du composant concerné :

- **Panne accidentelle (Crash) :** le composant soit, s'arrête complètement de fonctionner ou bien continue mais sans retourner à un état stable (valide).
- **Panne d'omission :** le composant n'est plus capable d'améliorer son service (échec total).
- **Panne de synchronisation (Timing) :** le composant effectue son traitement mais fournit le résultat en retard.
- **Panne Byzantine :** cette panne est de nature arbitraire ; le comportement du composant est donc imprévisible. Due à des attaques très malicieuses, ce type de pannes est considéré le plus difficile à gérer.

## 2.3 Définition de la tolérance aux pannes

Afin d'assurer la communication entre le nœud collecteur et les autres nœuds d'un réseau de capteurs, les protocoles de routage sont basés sur la communication multi sauts. Chaque nœud joue alors, en plus du rôle de source de données, le rôle d'un routeur. Toutefois, ces nœuds sont sujets à de nombreuses pannes, dues principalement à l'épuisement des batteries et aux destructions physiques (par exemple, suite à un écrasement par des animaux). Ainsi, la panne de nœuds entraîne la perte des liens de communication et donc un changement significatif dans la topologie globale du réseau. Ceci peut affecter d'une façon considérable la connectivité du réseau et diminuer, en conséquence, sa durée de vie.

La propriété de tolérance aux pannes est définie par l'aptitude du réseau à maintenir ses fonctionnalités, en cas de panne de certains de ses nœuds. Elle vise donc à minimiser l'influence de ces pannes sur la tâche globale du réseau [16].

## 2.4 Procédure générale de tolérance aux pannes

La conception d'une procédure pour la tolérance aux pannes dépend de l'architecture et des fonctionnalités du système. Cependant, certaines étapes générales sont exécutées dans la plupart des systèmes [17] comme c'est illustré dans la figure

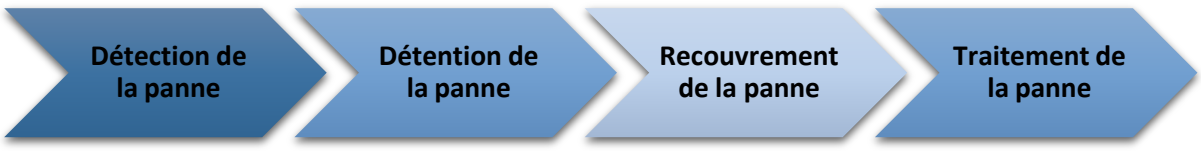



**Figure 2. 2 :** Procédure générale de tolérance aux pannes.

### 2.4.1 Détection de la panne

C'est la première phase dans chaque schéma de tolérance aux pannes, dans laquelle on reconnaît qu'un événement inattendu s'est produit. Les techniques de détection de pannes sont généralement classifiées en deux catégories : en ligne et autonome (offline). La détection offline est souvent réalisée à l'aide de programmes de diagnostic qui s'exécutent quand le système est inactif. La détection en ligne vise l'identification de pannes en temps réel et est effectuée simultanément avec l'activité du système.

### 2.4.2 Détention de la panne

Cette phase établit des limites des effets de la panne sur une zone particulière afin d'empêcher la contamination des autres régions. En cas de détection d'intrusion, par exemple, l'isolation des composants compromis minimise le risque d'attaque des composants encore fonctionnels.

### 2.4.3 Recouvrement de la panne

C'est la phase dans laquelle on effectue des opérations d'élimination des effets de pannes. Les deux techniques les plus utilisées sont « masquage de panne » qui utilise l'information redondante correcte pour éliminer l'impact de l'information erronée, et « répétition » qui effectue, après la détection d'une panne, un nouvel essai pour exécuter une partie du programme, dans l'espoir que la panne soit transitoire.

### 2.4.4 Traitement de la panne

Dans cette phase, la réparation du composant en panne est effectuée. La procédure de réparation dépend du type de la panne. Les pannes permanentes exigent une substitution du composant avec un autre composant fonctionnel.

## 2.5 Les techniques de gestion des pannes

### 2.5.1 Approches de redondance

L'approche la plus classique pour la tolérance aux fautes dans les réseaux de capteurs est la redondance [16]. Les réseaux de capteurs sont généralement denses et redondants. En effet, suivant l'application, on déploiera plus ou moins de capteurs dans un souci d'allongement de la durée de vie de l'application et l'amélioration de la tolérance aux pannes. Cette technique est très efficace mais également très coûteuse, car il faut prévoir plusieurs capteurs pour chacun des éléments à surveiller. Dans ce qui suit, nous décrivons quelques exemples de travaux utilisant les techniques de redondance dans les RCSF.

### 2.5.1.1 Redondance matérielle des nœuds

Certaines approches de redondance introduisent un mécanisme d'endormissement des nœuds [17] [18] afin de prolonger la durée de vie du réseau. Ce mécanisme nécessite néanmoins la mise en place de méthodes d'ordonnancement des activités. Une de ces approches est présentée dans [17]. Elle adopte une technique de détection d'un capteur défectueux et sa substitution par un nœud dans l'état endormi. La technique de détection consiste à diviser successivement la zone de captage en sous-zones afin de trouver le capteur défectueux. Au départ, on divise le réseau en quatre zones disjointes dont chacune a un nœud maître (celui qui a l'ID le plus élevé ou éventuellement un niveau d'énergie le plus élevé). Celui-ci se charge de calculer périodiquement le débit de sa propre zone. S'il détecte que le débit réel est inférieur à un seuil prédéterminé de la zone, il divise la zone en quatre quadrants. Le processus de division et de test de débit se répète dans chaque quadrant jusqu'à atteindre un quadrant qui ne contient qu'un seul nœud, considéré alors comme étant un nœud défaillant puisqu'il dégrade le débit de la zone. La technique de substitution choisit dans le voisinage du nœud suspect un nœud endormi en bon état (qui a un débit plus élevé que le nœud suspect à l'éveil). La simulation montre l'efficacité de l'algorithme à identifier plusieurs capteurs défectueux dans une même zone et à détecter les pannes dans plusieurs zones simultanément.

### 2.5.1.2 Redondance Matérielle – Sauvegarde des données

Une solution pour la tolérance aux pannes des nœuds capteurs et des puits est présentée dans [18]. L'objectif de l'algorithme est d'assurer une meilleure qualité des données avec une consommation minimale de l'énergie. L'algorithme suppose que les nœuds soient homogènes en termes de ressources donc n'importe quel capteur peut prendre le rôle du puits. De plus, il introduit un mécanisme d'endormissement pour prolonger la vie du réseau. Pour cela, dans chaque zone de captage, il suffit d'avoir un capteur source au minimum qui prend en charge la transmission des données au puits. L'algorithme utilise la technique du « data checkpointing » et la sauvegarde incrémentale de la mémoire du puits dans la mémoire d'un autre capteur. Celui-ci remplace le puits dans l'agrégation et la transmission des données à l'utilisateur final lorsque le puits tombe en panne. Pour cela, chaque nœud puits envoie à chaque période *(T*) à un nœud prédécesseur les données qu'il maintient. Si le niveau de la batterie du puits arrive à une valeur seuil, il envoie un message de contrôle au nœud prédécesseur indiquant qu'il est incapable de remplir les fonctions de puits. Si le niveau de la batterie de ce nœud est suffisant (supérieur à la valeur seuil), il prend le rôle du puits. Sinon, il envoie le message de contrôle à son tour au nœud qu'il le précède et ainsi de suite jusqu'à la fin du chemin de « checkpoint ». La longueur du chemin est un paramètre de l'algorithme et affecte considérablement la consommation d'énergie. Si la longueur du chemin est égale à trois, cela signifie que la sauvegarde se fait à trois niveaux.

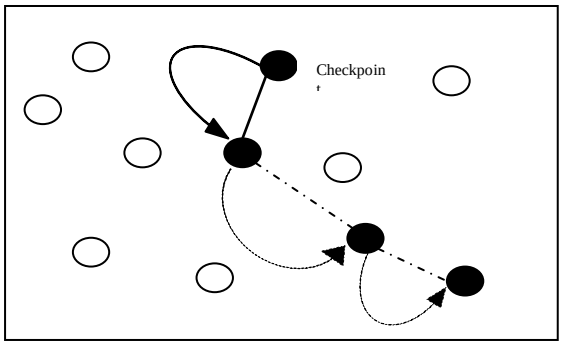



**Figure 2. 3 :** Chemin du « data checkpoint » du puits

## 2.5.2 Les approches de détection des pannes :

La détection de pannes est la première phase de gestion de pannes, où un échec inattendu devrait être correctement identifié. Les approches de détection de pannes dans les RCSF sont classées en trois catégories : approche centralisée, distribuée et l'approche par clustering.

### 2.5.2.1 Les approches centralisées de détection des pannes

Dans cette approche les nœuds malveillants ou en panne sont détectés par un nœud capteur géographiquement localisé. Ce nœud adopte un modèle de détection actif ou il injecte des messages dans le réseau pour localiser les nœuds échoués ou suspects. Les approches les plus communes de détection de pannes sont les suivantes :

- Sympathy [19] est un outil centralisé, dans lequel chaque nœud du réseau envoie périodiquement au puits des métriques qui sont de trois types : métriques de connectivité (table de routage, liste des voisins), métriques de flux (le nombre de paquets transmis et reçus par chaque nœud, le nombre de paquets échangés entre chaque nœud et le puits) et les métriques du nœud (temps de démarrage, le nombre de mauvais et de bons paquets reçus). Le puits collecte les métriques de manière passive en écoutant le trafic écoulé dans le réseau et de manière active car les nœuds envoient explicitement les métriques à chaque période T. La collecte n'est possible que pour les nœuds qui sont au prochain saut du puits. Pour cette raison, la collecte active des métriques est une obligation même si elle est très coûteuse en termes de communication et d'énergie. Une fois les métriques collectées, le puits peut identifier le type de(s) panne(s) produite(s*)* (franche, omission, synchronisation, transmission), ses principales causes (l'écrasement d'un capteur, le redémarrage du puits, l'absence de voisins, l'absence de route au puits, un mauvais chemin vers le nœud, un mauvais chemin au puits) et ses sources (le nœud lui-même, le réseau ou le puits). De ce fait, une panne est détectée quand un capteur génère un trafic inférieur à celui prévu. Ensuite, un arbre de décision est conçu pour faciliter l'analyse des causes des pannes (perte de connectivité, problèmes dans le routage, *etc*.). Par exemple, si le puits n'a pas reçu un paquet d'un nœud N au bout d'une durée donnée et si le nœud N n'est inclus dans la liste des voisins d'aucun autre nœud du réseau, N est supposé écrasé. Comme toute approche

centralisée, « Sympathy » induit une surcharge dans la communication (surcharge de 30% du trafic écoulé), ce qui limite le passage à l'échelle.

- Jessica Staddon et d'autres [20] proposent un algorithme centralisé de détection des pannes dans lequel chaque nœud envoi des informations sur sa topologie à la station de base (liste de ses nœuds voisins), cette dernière utilise ces informations pour construire la topologie intégrale du réseau. Une fois la topologie construite, les nœuds malveillants ou en panne peuvent être efficacement tracés en utilisant une stratégie de découpage en zone. Notons que cette approche suppose que chaque nœud possède un numéro d'identification unique et la station de base est apte à transmettre directement des messages vers n'importe quel nœud du réseau. Cela nécessite généralement un envoi supplémentaire de messages vers les nœuds, et il est par conséquent très coûteux. De plus, cette approche n'est pas applicable aux réseaux de capteurs orientés événement qui envoient des messages uniquement quand il ya un événement dans le réseau.

- Sapon Tanachaiwiwat et d'autres [21] proposent un algorithme dans lequel la station de base utilise des paquets marqués pour identifier les nœuds malveillants ou en panne. Quand le nœud ne répond pas à un message envoyé par la station de base ou quand un ensemble de paquets est dropé cela dit que le nœud est en panne ou qu'il est malveillant.

Bien que l'approche centralisée est efficace pour la détection des pannes mais elle est très coûteuse en termes de consommation d'énergie. De plus, elle devient inefficace lorsqu'un grand nombre de capteurs doit être déployé.

### 2.5.2.2 Les approches distribuées de détection des pannes :

Cette approche encourage le concept de prise de décision local qui distribue la gestion des pannes dans le réseau et décharge le nœud central qui n'est informé qu'en cas de panne. Les approches distribuées les plus communes de détection de pannes sont les suivantes :

- **Auto détection**

Dans [22] une approche distribuée de détection des pannes a été proposé, elle permet à un nœud d'exécuter un auto-diagnostique basé sur les mesures des accéléromètres qui détermine si le nœud aura un dysfonctionnement du matériel.

Dans [23] une approche distribuée a été proposé dans laquelle la défaillance d'un nœud est détectée en comparaison avec un modèle prédéfini.

- **Coordination entre voisins**

Les nœuds capteurs coordonnent avec leurs voisins pour détecter et identifier la panne dans le réseau avant d'informer le nœud central ce qui réduit la charge du réseau et par conséquent réduit la consommation d'énergie, il existe beaucoup de développements d'une telle approche.

Dans [24] [25] [26] [27], l'idée consiste à comparer les mesures de capteurs similaires qui se trouvent dans une même zone géographique. Tant que les valeurs délivrées par ces capteurs restent égales entre elles, l'information est considérée comme fiable car il est très improbable que tous les capteurs fassent la même erreur de mesure au même instant. Si une de ces valeurs s'écarte significativement des autres, c'est qu'un problème est apparu sur le capteur qui délivre cette valeur.

Dans [25], l'algorithme identifie tout d'abord les capteurs qui sont en bon état, ce après une suite d'échanges de valeurs, puis utilise ces capteurs comme références pour tester les autres capteurs. La simulation montre que l'algorithme induit une surcharge assez élevée en termes de communication et d'énergie. Pour réduire la surcharge du réseau, une architecture arborescente est adoptée dans [27]. Après un certain nombre d'échanges, un capteur est considéré en bon état. Il sert alors de référence pour tester les capteurs de son sous-arbre.

Dans [28] un algorithme a été développé qui n'exige pas la connaissance de la position du nœud et qui fonctionne même quand la moitié des nœuds sont défectueux, dans cet algorithme les meilleurs résultats captés sont utilisés pour identifier les nœuds suspects.

### 2.5.2.3 Approches par cluster pour la détection des pannes

Le « *clustering* » est devenue une technologie émergente pour construire des applications évolutives pour les réseaux de capteurs sans fil [28]. Ann T.Tai [29] tire une solution de détection de pannes efficace utilisant une hiérarchie de communication à base de cluster comme dans la figure 2.4

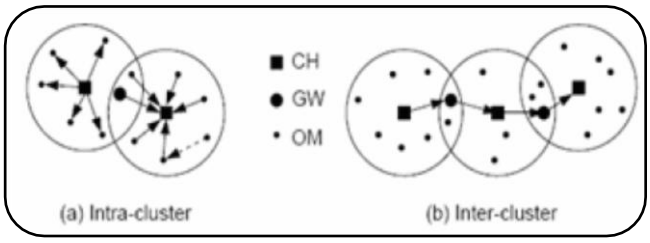



**Figure 2. 4 :** Diffusion des messages intra-cluster et propagation de l'information inter-cluster.

La détection par *intra-cluster* est adoptée pour identifier les nœuds défectueux dans chaque groupe, le *cluster head* détecte les nœuds suspects en échangeant des messages avec les voisins qui sont à un saut, ensuite il identifie les nœuds défectueux selon une règle de détection de pannes, après identification, il propage cette information à tous les groupes.

Venkataraman [31] a développé un algorithme de détection et de recouvrement de pannes dans lequel le nœud doit envoyer un « *fail-report-msg »* à ses voisins quand son énergie diminue au delà d'un certain seuil pour qu'ils initient la procédure de recouvrement de la panne et pour qu'ils restent connectés au cluster.

Akbari et d'autres [30] ont développés un algorithme qui donne des résultats meilleurs que ceux donné par Venkataraman et ceux de Gupta [31]. Cet algorithme à un mécanisme de détection de pannes dans lequel un 2$^{ème}$ cluster head est élu, quand l'énergie du cluster head diminue au delà d'un certain seuil il envoi un message à tous les nœuds de son cluster incluant le cluster head secondaire, ce message est considéré comme un signe pour le cluster head secondaire.

## 2.5.3 Le diagnostic des pannes :

### 2.5.3.1 Classification des approches de diagnotic des pannes

De nombreuses approches ont été développées en vue du diagnostic des défaillances dans le système à base de RCSF. Elles se distinguent par les mécanismes de la collecte des informations du diagnostic et la structure de prise de décision concernant la détection.

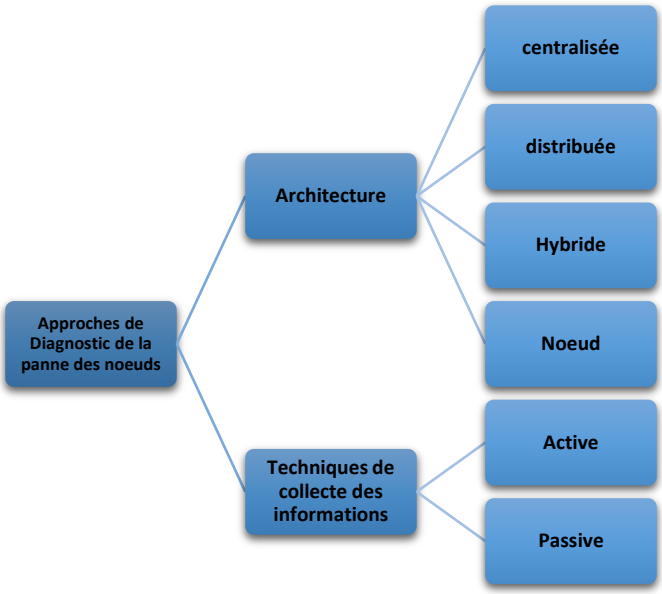



**Figure 2. 5 :** classification des approches de diagnostic des pannes

#### a. Classification selon l'architecture :

Selon la structure de la prise de décision et le(s) entité(s) intervenante(s) dans le processus de détection et de diagnostic des défaillances, les auteurs classifient les approches de diagnostic en plusieurs catégories : approche centralisée, approche distribuée, approche hybride et approche au niveau du nœud.

- **Approche centralisée :** Elle repose sur un nœud central qui diagnostique les fautes en se basant sur des informations recueillies des nœuds du réseau. Cette approche assure une supervision globale de l'état du réseau permettant de garantir une exactitude quant au diagnostic des problèmes complexes (par exemple, la défaillance des nœuds critiques). Les

limites de cette approche sont liées à une dépendance forte due à la centralisation exclusive des opérations de diagnostic. D'une part, cette approche n'est pas robuste vis-à-vis de la perte des messages transmis sur un réseau sans fil de communication multi saut. De plus, l'arrêt accidentel du point central entraîne également l'arrêt du système de diagnostic. D'autre part, elle est coûteuse en termes de consommation d'énergie, et ne permet pas le passage à l'échelle. L'approche centralisée favorise principalement les réseaux de petite taille.

- **Approche distribuée :** La détection des pannes est réalisée de façon distribuée par l'ensemble des nœuds du réseau. Elle permet d'améliorer la robustesse du système. Cependant, des mécanismes de coopération doivent être mis en œuvre pour assurer la cohérence dans les opérations de gestion exécutées par les nœuds.

- **Approche hybride :** Elle combine le principe des deux approches centralisée et distribuée. Le but est de pouvoir profiter des avantages des deux approches : l'exactitude et la précision des approches centralisées et l'efficacité de la gestion de l'énergie et le passage à l'échelle des approches distribuées.

- **Approche de diagnostic au niveau du nœud :** Le diagnostic se base sur des informations sur l'état d'un nœud sans tenir compte du comportement des autres nœuds du réseau. Cette approche ne permet pas le diagnostic des fautes réseaux.

| Architecture | Centralisée | Distribuée | Hybride |
|---|---|---|---|
| **Précision/Exactitude** | Bonne | Moyenne | Bonne |
| **Consommation d'énergie** | Forte | Faible | Moyenne |
| **Passage à l'échelle** | Non | Oui | Oui |
| **Robustesse** | Non | Oui | Oui |
| **Latence de détection** | Bonne | Faible | Faible |
| **Impacts sur la performance du réseau** | Oui | Faible ou moyenne | Moyenne |
| **Complexité d'implémentation** | Non | Oui | Oui |

Table 2. 1 : Comparaison entre les approches centralisées, distribuées et hybrides

## b. Classification selon les techniques de collecte des informations

L'outil de diagnostic récupère les informations sur le système surveillé selon deux approches principales [24]: active et/ou passive.

- **Approche active :** Les nœuds génèrent des paquets spécifiques au diagnostic (les paquets de contrôle). La transmission des paquets s'effectue en utilisant le canal de communication principal de l'application. L'approche active est coûteuse en termes de consommation de ressources. Pour cela, certaines solutions introduisent des nœuds « mobiles » dotés de batteries plus puissantes et capables de se déplacer pour récupérer les informations des nœuds du réseau. Cependant, elle ne nécessite pas de matériel supplémentaire pour transmettre les paquets de contrôle.

- **Approche passive :** L'approche passive ne requiert aucune intervention de la part des nœuds du réseau, et ne génère pas de paquets supplémentaires dans le réseau. Elle peut être réalisée par le puits, ou par les nœuds du réseau. Un avantage principal du diagnostic passif est la transparence. Le diagnostic s'effectue sans aucune interférence avec les opérations normales du réseau. Ce qui permet de garder les ressources des nœuds du réseau et la performance globale du système. Un autre avantage est le support d'une grande diversité de plateformes.

  Un autre modèle de l'approche passive, est l'approche de marquage des paquets « piggybacking ». Il consiste à insérer les informations de diagnostic dans les paquets de données. Il permet de réduire la consommation de la bande passante, en évitant d'injecter des paquets supplémentaires dans le réseau. L'inconvénient principal du marquage des paquets, est que la taille des informations à insérer est limitée à la taille maximale des paquets.

| **Approche de la collecte d'information** | **Active** | **Passive** | **Marquage** |
|---|---|---|---|
| **Transparence** | Non | Oui | Oui |
| **Coût mémoire** | Oui | Non | Non |
| **Consommation d'énergie** | Forte | / | Faible |
| **Fiabilité** | Faible | élevée | Faible |
| **Support de la mobilité** | Oui | Non | Oui |

| | | | |
|---|---|---|---|
| **Passage a l'échelle** | / | Oui | Oui |
| **Domaines d'applications** | Tout type d'application | Application de la collecte des données | Application de la collecte des données |

Table 2. 2 : Mécanismes de transmission des informations du diagnostic

### c. Quelques exemples d'approches de diagnostic des pannes :

Les travaux dans [34,35] supposent que la partie logicielle du capteur est déjà tolérante aux pannes et que la majorité des pannes sont des pannes matérielles. Quand a Thomas [33], il suppose que la majorité des pannes sont dues à l'environnement dans lequel les capteurs sont déployés.

## 2.5.4 Les approches de recouvrement des pannes des nœuds

Le recouvrement de la panne est l'étape dans laquelle le réseau de capteur est reconfiguré et reconstitué d'une telle façon que les pannes n'influent pas sur les performances du réseau, les approches existantes isolent les nœuds défectueux. Selon Marti [34], après détection de la panne, le nœud capteur doit choisir un autre voisin pour acheminer ses paquets. Tandis que Win MS [35] a proposé que le nœud central détecte la région faible du réseau (exemple, celle qui a une faible énergie) en comparant l'état actuel du réseau avec un modèle historique donnant des informations sur le réseau (exemple, schéma d'énergie) , après détection il règle les nœuds de cette région de façon à ce qu'ils envoient les informations moins fréquemment qu'avant afin de conserver leur énergie et maximiser ainsi leur durée de vie. Dans [31], quand un nœud passerelle tombe en panne, tous les nœuds du groupe sont réattribués à d'autres passerelles en bon état. Ceci consomme plus de temps vu que tous les membres du groupe sont impliqués dans le processus de rétablissement.

# 2.6 Conclusion

La tolérance aux pannes dans un réseau de capteurs est la capacité de ce dernier à maintenir son bon fonctionnement malgré la présence de quelques défaillances. Ces défaillances peuvent survenir par manque d'énergie ou en raison de dommages physiques ou d'interférences environnementales. En effet, la panne de quelques nœuds entraine la perte des liens de communication et ainsi un changement significatif dans la topologie du réseau.

Dans ce chapitre, nous avons présenté toutes les approches existantes de gestion des pannes dans les RCSF. Nous les avons classifiées en trois phases qui sont la détection, le diagnostic et le recouvrement de pannes.

Dans le chapitre suivant nous présentons les différents protocoles qui assurent la tolérance aux pannes dans les réseaux de capteurs sans fil.

CHAPITRE 3

# LES PROTOCOLES DE ROUTAGE TOLERANTS AUX PANNES DANS LES RESEAUX DE CAPTEURS SANS FIL

## SOMMAIRE

# 3.1 Introduction

La défaillance des nœuds dans un réseau de capteurs peut être engendrée par plusieurs causes, notamment l'épuisement d'énergie, l'endommagement physique, ou les interférences liées à l'environnement.

La propriété de tolérance aux pannes est définie par l'habilité du réseau à maintenir ses fonctionnalités sans interruptions provoquées par la panne des capteurs. Elle vise donc à minimiser l'influence de ces pannes sur la tâche globale du réseau [16]. L'approche la plus célèbre de tolérance aux pannes est le routage multi-chemins, où plusieurs chemins multiples entre les nœuds source et la station de base sont déterminés au détriment de la consommation d'énergie accrue et la génération du trafic.

Dans ce chapitre, nous tenons à présenter une synthèse de quelques protocoles de routage tolérants aux pannes proposés dans la littérature.

# 3.2 Classification des protocoles de tolérance aux pannes dans les RCSF

Les protocoles tolérants aux pannes peuvent être vus de plusieurs angles différents. De ce fait, un ensemble de critères est défini pour les classifier. Nous citons, entre autre, deux principales catégories ; à savoir les classifications temporelles et architecturales

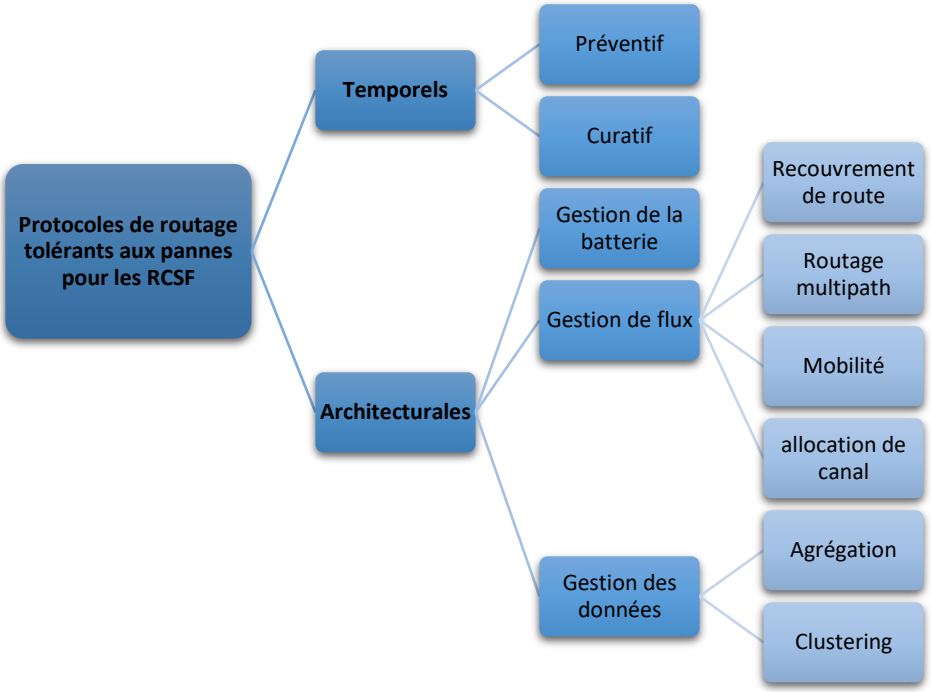



**Figure 3. 1** Classification des protocoles de tolérance aux pannes

## 3.2.1 Classification temporelle

Dans la classification temporelle, nous divisons l'ensemble des algorithmes en deux catégories, et cela selon la phase de traitement. Si le traitement est effectué avant la panne ; on parle donc d'algorithmes préventifs. Sinon, les algorithmes sont dits curatifs.

- **Algorithme préventif :** Implémente des techniques tolérantes aux pannes qui tentent de retarder ou d'éviter tout type d'erreur afin de garder le réseau fonctionnel le plus longtemps possible. La conservation d'énergie à titre d'exemple, permet de consommer moins d'énergie et évite donc une extinction prématurée de la batterie ce qui augmente la durée de vie des nœuds.

- **Algorithme curatif :** Utilise une approche optimiste, où le mécanisme de tolérance aux pannes implémenté n'est exécuté qu'après la détection de pannes. Pour cela, plusieurs algorithmes de recouvrement après pannes sont proposés dans la littérature, par exemple : le recouvrement du chemin de routage, l'élection d'un nouvel agrégateur, etc.

## 3.2.2 Classification architecturale

Cette classification traite les différents types de gestion des composants, soit au niveau du capteur individuellement ou bien sur tout le réseau. Nous distinguons trois catégories principales :

- **Gestion de la batterie :** Cette catégorie est considérée comme une approche préventive, où les protocoles définissent une distribution uniforme pour la dissipation d'énergie entre les différents nœuds capteurs ; afin de mieux gérer la consommation d'énergie et augmenter ainsi la durée de vie de tout le réseau. En outre, le mécanisme de mise en veille est une technique de gestion de batterie. En effet, les protocoles déterminent des délais de mise en veille des nœuds capteurs inactifs pour une meilleure conservation d'énergie.

- **Gestion de flux :** Cette catégorie regroupe les techniques qui définissent des protocoles de gestion de transfert des données (routage, sélection de canal de transmission, etc.). Nous pouvons trouver des approches préventives ou curatives sur les différentes couches (réseau, liaison de données, etc.) telles que :

    - **Routage multipath :** utilise un algorithme préventif pour déterminer plusieurs chemins depuis chaque capteur vers le nœud collecteur. Ceci garantit la présence de plus d'un chemin fiable pour la transmission et offre une reprise rapide du transfert en cas de panne sur le chemin principal et choisissant un des chemins qui restent.

- **Recouvrement de la route :** après détection de panne, une technique curative permet de créer un nouveau chemin qui soit le plus fiable pour retransmettre les données.
- **Allocation du canal :** cette solution est implémentée au niveau de la couche MAC. Elle permet d'effectuer une allocation du canal de transmission d'une manère à diminuer les interférences entre les nœuds voisins et éviter les collisions durant le transfert.
- **Mobilité:** certains protocoles proposent comme solution tolérante aux pannes la sélection d'un ensemble de nœuds mobiles chargés de se déplacer entre les capteurs et collecter les données captées. Ceci réduira l'énergie consommée au niveau de chaque capteur en éliminant sa tâche de transmission. Un nœud mobile est généralement doté d'une batterie plus importante que celle d'un nœud capteur.

- **Gestion des données :** Les protocoles classés dans cette catégorie offrent une meilleure gestion de données et de leur traitement. Deux principales sous-catégories sont déterminées :

  - **Agrégation :** considérée comme approche préventive, l'opération d'agrégation effectue un traitement supplémentaire sur les données brutes captées depuis l'environnement. Un nœud agrégateur combine les données provenant de plusieurs nœuds en une information significative. Ce qui réduit considérablement la quantité de données transmises en consommant moins d'énergie pour leur dissémination. Ceci permet donc d'augmenter la durée de vie du réseau.
  - **Clustering :** une des importantes approches pour traiter la structure d'un réseau de capteurs est le clustering. Il permet la formation d'un backbone virtuel qui améliore l'utilisation des ressources rares telles que la bande passante et l'énergie. Par ailleurs, le clustering aide à réaliser du multiplexage entre différents clusters. En outre, il améliore les performances des algorithmes de routage. Plusieurs protocoles utilisent cette approche préventive et parfois elle est considérée comme une approche curative.

## 3.3 Les protocoles de routage tolérants aux pannes dans les RCSF

Les protocoles de routage proposés pour les réseaux de capteurs pour assurer la tolérance aux pannes permettent une fiabilité de délivrance de paquets à la station de base même en cas de panne de certains nœuds, cette dernière est traitée au niveau de la couche réseau. Dans ce qui suit, nous présentons les fonctionnalités de certains protocoles de routage tolérants aux pannes et nous discutons leurs limites

### 3.3.1 RERP (Routing with Error Reporting Protocol)

Dans RERP [36], il est supposé que chaque nœud a au moins deux voisins dans la direction vers la station de base. La capacité d'un nœud tolérant aux pannes dépend du nombre des nœuds voisins actifs. Le protocole RERP est un protocole de routage proactif et sa conception comporte deux tâches :

- **Mise en place de RERP :** cette tâche s'exécute en cinq phases :

  - **Phase d'avertissement :** Dans cette phase, la station de base diffuse un paquet d'avertissement à ses nœuds voisins pour indiquer qu'elle peut recevoir des paquets de données. Les nœuds qui reçoivent ce paquet enregistrent le chemin vers la station de base dans la table de routage.
  - **Phase d'initialisation :** Lors de cette phase, les nœuds qui n'ont pas de chemin direct vers la station de base diffusent une requête de découverte de route (RREQ) vers la station de base. Quand un concentrateur reçoit ce paquet, il diffuse une réponse (RREP) s'il existe un chemin entre lui et le concentrateur, sinon le paquet (RREQ) sera ignoré.
  - **Sélection de la route :** Elle est faite sur la base de l'énergie c'est-à-dire que s'il y'a deux routes qui mènent vers la destination celle qui a le plus grand degré d'énergie est choisie.
  - **Phase de transfert de données :** Les nœuds capteurs génèrent des paquets de données à chaque fois qu'ils détectent toute nouvelle information. Cette information est transmise à la station de base dans un mode multi-sauts.
  - **Table de secours :** En plus du chemin principal, un chemin alternatif est prévu pour tous les nœuds du réseau. Chaque fois qu'un nœud reçoit un paquet RREP, s'il ne dispose pas de chemin direct vers la station de base, il stocke le chemin dans la table de routage, et il stocke les paquets (RREP) dans une table de secours. La table de routage de secours dispose de deux champs, l'identifiant du nœud ID et son énergie.

- **Rapport d'erreurs :** Les fonctions pour reporter les erreurs sont incorporées dans ce protocole de routage dans lesquelles figurent les types de messages suivants :

  - **Echec des liens :** Il est généré dans les deux cas. Le premier se produit quand un RTS est envoyé mais aucune CTS correspondant n'est reçu et le nombre maximal de tentatives est dépassé. Le second se passe quand un paquet de données a été transmis, mais il n'a jamais reçu un ACK de réception et le nombre maximal de tentatives est dépassé.
  - **Message de batterie critique :** Ce message est généré lorsque le niveau de la batterie d'un nœud est inférieur à un seuil critique. Ce message est envoyé au nœud source qui a envoyé les données et également aux voisins de ce nœud. Quand les autres nœuds

reçoivent ce message ils suppriment l'identifiant du nœud défaillant de leurs tables de routage ou de leur table de voisins.

- **Message de destination inaccessible :** Ce message est généré lorsque le paquet de données n'est pas transmis au nœud de destination en raison de l'indisponibilité du chemin.

- **Sélection du chemin de secours :** Chaque nœud possède une table de routage de secours dans laquelle il stocke un chemin de secours vers la destination. Quand un nœud échoue dans la transmission de paquet de donnée, alors son voisin consulte la table de secours pour trouver le chemin alternatif afin qu'il puisse transmettre le paquet de données.

Dans RERP la communication entre les nœuds est réalisée par des messages Requête/Réponse où le nœud expéditeur envoie une requête « Hello » au nœud de destination. Ce type de messages est utilisé pour vérifier si le voisin est accessible et pour calculer le temps de parcours.

RERP présente certaines limitations telles que la consommation d'énergie qui est assez grande lors de la diffusion des rapports d'erreurs.

### 3.3.2 ENFAT-AODV (The Enhanced Fault-Tolerance Mechanism of AODV Routing Protocol)

The Enhanced Fault-Tolerance Mechanism of AODV Routing Protocol (ENFAT-AODV) [37] est un protocole de routage tolérant aux pannes qui offre une mise en place d'itinéraire rapide et efficace entre les nœuds.

En outre, ENFAT-AODV permet aux nœuds sur une voie principale de transmettre des données pour obtenir un chemin de secours, qui est utilisé lorsque le chemin principal sera perdu, pour établir un lien entre les nœuds dans les meilleurs délais. Le nombre de sauts est utilisé en tant que métrique pour sélectionner le chemin. Si de multiples RREP sont reçus par la source, le chemin le plus court est choisi.

Toutefois, pour ENFAT-AODV, certains champs sont ajoutés dans les paquets de contrôle tels que "BACKUP" drapeau (en RREQ et RREP), "UPDATE" drapeau (en RREQ) et le champ "Distance pour Dest" (dans RREQ). En outre, ENFAT-AODV nécessite que certains messages (par exemple, RREQ) doivent être largement diffusés, peut-être à travers le réseau. La zone de diffusion de ces RREQs est indiquée par le TTL dans l'en-tête IP. ENFAT-AODV se déroule comme suit :

- **Découverte de la route principale :** Quand un chemin principal de livraison de données vers la station de base est nécessaire, le nœud source diffuse un message de découverte d'une

route principal (RREQ) à la station de base. Si le nœud ne connaît pas la route principale menant à la destination, il transmet le RREQ à ses voisins. Sinon s'il est la destination ou bien s'il connaît la route principale menant à la destination, il va générer un itinéraire (principale RREP).

Comme le RREP principale est renvoyé à la source, chaque nœud intermédiaire qui traite le RREP principale crée un chemin principal vers la station de base. Lorsque la source reçoit le RREP principale, elle enregistre la route principale menant à la destination dans sa table de routage principale.

- **Construction de la route de secours :** les nœuds d'un chemin principal qui reçoivent un RREP principale créent un chemin de sauvegarde vers la station de base en diffusant un paquet de sauvegarde RREQ. Après la diffusion du RREQ, le nœud attend un paquet RREP de la destination ou d'un nœud intermédiaire qui peut satisfaire les conditions spécifiées comme suit :

  - Il dispose d'une entrée de sauvegarde active du chemin principal vers la station de base.
  - Il n'est pas un nœud sur le chemin principal.
  - Et le nombre de sauts du chemin de sauvegarde active à partir du nœud intermédiaire à la destination est minimal, et ce pour garantir qu'il fournira un chemin de sauvegarde court.

- **Entretien de la route :** Pendant la période de livraison des paquets de données, lorsque le chemin principal n'est pas valide ou reçoit un paquet de données destiné au nœud de destination pour laquelle il ne dispose pas d'un chemin actif principal, le nœud utilise immédiatement sa route de secours pour livrer les prochains paquets de données sans interruption. Par la suite, le nœud sur le nouveau chemin principal, qui utilise une route de secours, dirige un processus "Découverte de route de secours" visant à trouver une voie nouvelle. Ce qui augmente de plus la fiabilité et la disponibilité par rapport au protocole de routage AODV.

La limitation de ce protocole repose sur la consommation de l'énergie à cause de l'inondation par des messages de contrôle.

### 3.3.3 IHR (Informer Homed Routing)

IHR [38] est un protocole de routage tolérant aux pannes qui a été créé en se basant sur le protocole DHR [39] qui se déroule comme suit :

- **Formation des groupes et élection des chefs de groupes principaux :** Dans cette phase les chefs de groupes sont élus sur la base de l'énergie, après élection ils envoient leurs statuts aux nœuds ou chaque nœud choisi le chef de groupe le plus proche pour le rejoindre.
- **Election des chefs de groupes de secours :** Après formation des groupes, le chef de groupe choisi parmi les nœuds de son groupe celui qui a le plus grand degré d'énergie et le sélectionne comme chef de groupe de secours (BCH) dont le rôle est d'acheminer les données vers la station de base dans le cas ou le chef de groupe principale tombe en panne. Après l'élection du chef de groupe de secours, le chef de groupe principale envoi cette information à tous les nœuds de son groupe.

A chaque période de temps le chef de groupe de secours envoi un message «Beacon » au chef de groupe principale, si ce dernier n'est pas en panne il lui répond. Sinon le chef de groupe de secours attend trois round, si au bout de ce temps il ne reçoit aucune réponse il déclare le nœud comme en panne et envoi un message aux nœuds de son cluster pour qu'ils lui communiquent leurs données.

## 3.3.4 CFS (Cluster based Fault tolerant Scheme)

CFS [40] est un protocole de routage tolérant aux pannes qui se déroule en deux phases :

- **Formation des clusters :**

Lors de la première phase qui est la phase de formation des clusters et l'élection des clusters head, deux clusters sont élus : un cluster head primaire ($CH_p$) et un vice cluster head ($CH_v$), ces deux clusters head sont élus sur la base du poids du capteur tel que le poids est une combinaison entre l'énergie résiduelle du capteur avec sa densité qui se calcule comme suit :

$$poids(u) = \propto * p_{2(u)} + \beta * E(u)^{\wedge} \propto + \beta = 1$$

Les valeurs de $\propto et\ \beta$ sont choisis en fonction de l'application par exemple si on veut favoriser les nœuds qui ont le plus grand degré d'énergie, on doit attribuer une très grande valeur à $\beta$.

Le paramètre du poids est calculé au début de chaque Round afin d'éviter l'épuisement rapide de l'énergie du capteur. En faite, chaque nœud calcule son poids puis envoi un message «Hello » à tous ses voisins incluant trois champs additionnels : poids, ($CH_p$) et ($CH_v$) ou($CH_p$) et ($CH_v$) sont initialisés à zéro. Après cet échange le nœud qui a le plus grand poids est élu comme ($CH_p$) et celui qui a le 2ème plus grand poids est élu comme ($CH_v$). Ensuite, chaque nœud met à jour son vecteur d'état en assignant aux champs($CH_p$) et($CH_v$) les identifiants correspondants à chacun d'eux.

Par la suite, le($CH_p$) envoi un message « ADV_CH » à tous ses voisins pour les inviter à le rejoindre. Chaque nœud qui reçoit ce message et qui n'est pas membre d'un autre groupe lui répond avec un message « REQ_JOIN », le($CH_p$) vérifie ensuite que la taille de son cluster n'a pas atteint ($Thresh_{upper}$) ou ($Thresh_{upper}$) représente la valeur seuil de la taille du cluster qui se calcule comme suit:

$$Thresh_{upper} = \frac{(\delta_2(u) + AVG)}{2} \quad \text{Ou} \quad \begin{cases} AVG = \\ \delta_2(u_i) = Max\{\delta_2(u_i);\ u_i \in v\} \end{cases} \frac{\sum_{i=1}^{n} \delta_2(u_i)}{n}$$

Ou :

$\delta_2(u_i)$: Permet d'avoir le nœud qui a le plus grand degré d'énergie.

$AVG$ : représente la valeur moyenne du degré de tous les nœuds du réseau.

Si c'est le cas il envoi au nœud un message « ACCEPT_CH ».

- **Construction des chemins de routage et envoi des données :**

Les CHs construisent des chemins de routage CH-à-CH qui seront utilisés pour la transmission des données. Chaque CH utilise ces chemins pendant l'intervalle de temps qui lui a été attribué.

Quand un nœud détecte un événement, il envoi un paquet de données au $(CH_p)$ qui se charge de les agréger et les communiquer à la station de base via une transmission directe ou multi-sauts, si ce dernier n'envoi pas d'acquittement, le $(CH_v)$ attend une période de temps puis envoi un ACK au nœud et considère que le $(CH_p)$ est en panne. Une fois les données envoyées au cluster head tous les nœuds du cluster éteignent leur radio jusqu'au prochain intervalle de temps et ce afin de conserver leur énergie.

## 3.3.5 FTCP-MWSN (Energy Efficient and Fault Tolerant Protocol for Mobile Wireless Sensor Network)

FTCP-MWSN [41] est un protocole de routage tolérant aux pannes conçu pour les réseaux de capteurs mobiles qui assume que tous les nœuds sont homogènes en termes de mobilité et se déplacent avec une vitesse constante et qu'à chaque fois qu'un nœud quitte le cluster un autre entre à ce même cluster. Il se déroule comme suit :

- **Formation des clusters et élection des Cluster Head:**

La station de base forme les clusters en se basant sur la position des capteurs et sélectionne les CHs en se basant sur le degré d'énergie et la position des CHs. Puisque les nœuds ont initialement le même degré d'énergie, le CH est élu en se basant sur la génération d'une valeur aléatoire comprise entre 0 et 1 de la même manière que celle du protocole LEACH. Une fois le CH élu il envoi un message à tous les nœuds contenant sa position ainsi que son identificateur. Chaque nœud qui reçoit ce message calcule la distance qui le sépare du CH lorsqu'elle est minimale, le nœud répond avec un message contenant sa position et son identificateur, lorsque le CH reçoit ce message il enregistre le nœud et envoi toutes les informations concernant son cluster à la station de base pour les opérations de contrôle centralisées.

- **La phase de transmission :**

Lors de cette phase le CH alloue à chaque nœud un intervalle de temps en utilisant la méthode TDMA, pendant lequel il doit envoyer les données au CH et calculé son mouvement à l'intérieur et à l'extérieur du cluster. Lorsque le CH reçoit ces données, il envoi un acquittement. Cependant, quand un nœud se déplace vers un nouveau cluster il doit envoyer un message «JOIN-REQUEST » au nouveau CH qui ne lui répond que lorsqu'un intervalle de temps se libère.

En effet, le CH notifie les nœuds de l'événement désiré exemple : envoyer un paquet de données lorsque la température dépasse 70°. Quand le nœud détecte l'événement pendant son intervalle de temps il envoi le paquet de données au CH sinon il envoi un petit paquet pour signaler au CH qu'il n'est pas en panne et qu'il ne s'est pas déplacé en dehors du cluster. Une fois le paquet de données ou le petit paquet reçu le CH renvoi un acquittement. Par contre s'il ne reçoit rien il déclare le nœud à la SB comme hors cluster et le supprime de la liste de ses membres de cluster et supprime également son intervalle de temps de son plan TDMA. De même lorsque le nœud ne reçoit pas d'acquittement il déduit

qu'il n'est plus attaché à ce CH à cause de la mobilité, ainsi il envoi un message « JOIN-REQUEST » à tous les CHs qui sont à sa portée.

Bien que ce protocole semble donner de bons résultats par rapport à d'autres protocoles mais il reste incomplet vu que Il n'a pas traité le problème de panne des CHs chose qui peut engendrer une perte des paquets de données et par conséquent provoquer une isolation d'une zone du réseau. De plus, il semble coûteux en terme d'énergie vu qu'il y'a beaucoup d'échange de message entre les nœuds et beaucoup de calculs qui se font à chaque fois. Ce protocole semble coûteux en termes de coût également vu qu'il nécessite à chaque fois la localisation de tous les nœuds.

## 3.3.6 MRP (Multilevel Routing Protocol)

MRP [42] est un protocole hiérarchique basé sur l'organisation des nœuds au sein du cluster avec une communication multi sauts inter-cluster qui se déroule en deux phases :

a. **La phase d'initialisation :** Lors de cette phase, les CHs sont élus et les groupes sont formés comme suit : chaque nœud calcule la probabilité $P_i(t)$ qui lui permet de savoir s'il est CH pour ce ROUND et ce en générant une valeur i comprise entre 0 et 1. Si cette valeur est inférieure à$P_i(t)$, le nœud est élu comme CH. Elle se décompose en plusieurs sous phases

   - **La phase d'organisation des clusters :** C'est la phase de formation des clusters ou chaque CH envoi en multicast un paquet « ADV » contenant l'identificateur du CH en utilisant CSMA/CA pour éviter les collisions entre les CHs. Chaque nœud qui reçoit ce message répond avec un paquet « JOIN » au CH le plus proche sur la base de la force du signal.
   - **La phase d'ordonnancement :** elle se décompose en deux sous phases, lors de la première phase le CH envoi un paquet « ACK » aux nœuds qui ont envoyés un paquet « JOIN » et ce afin de déterminer la liste des nœuds de son cluster en utilisant TDMA. Lors de la deuxième phase les rayons de chaque niveau sont calculés ce qui permet de déterminer le niveau auquel appartient chaque CH.

b. **La phase de transmission :** Dans cette phase, les données collectées depuis chaque nœud sont envoyées au CH qui les agrège et les communique au CH de niveau supérieur et ce jusqu'à ce que les données arrivent à la station de base.

### 3.3.6.1 IFTMRP (Inter-cluster Fault Tolerant Multilevel Routing Protocol)

L'objectif de ce protocole est de prendre en considération les pannes du CH lors de l'émission et de la réception et ce en assurant une tolérance aux pannes inter-cluster. Pour se faire chaque CH doit choisir deux CH voisins au lieu d'un seul et ce afin de pouvoir récupérer les données lorsque le 1$^{er}$ CH tombe en panne. Lorsque le CH envoi les données à son CH voisin, ce dernier doit lui envoyer un paquet « ACK », si le CH émetteur ne reçoit pas d'ACK, il renvoit les données au 2$^{ème}$ CH.

La limite de ce protocole est que les données collectées par les nœuds du cluster ou le CH est en panne sont perdues

### 3.3.6.2 IIFTMRP (Inter-Intra cluster Fault Tolerant Multilevel Routing Protocol)

Pour résoudre le problème de perte des données au sein du cluster, le protocole IIFTMRP a été développé. Dans IIFTMRP, le CH choisi parmi les membres de son cluster celui qui a le plus grand degré d'énergie comme vice CH qui est chargé d'envoyer les données lorsque le CH tombe en panne.

Pendant le slot réservé au vice CH, si le CH ne reçoit pas les données, il envoi un message d'erreur au vice CH pour le notifier qu'il est en panne. Une fois le message reçu, le vice CH devient responsable de l'émission des données vers les CH de niveau supérieure. Cette solution a résolu le problème posé mais elle reste coûteuse en termes d'énergie vu le nombre important de messages envoyés et reçus par les nœuds.

## 3.3.7 Protocole de routage dynamique tolérant aux pannes pour prolonger la durée de vie dans les RCSF

L'objectif de ce protocole est de maintenir la connectivité du réseau, même si un nœud est sur le point d'épuiser son énergie pour assurer la livraison de données à la station de base tout en prolongeant la durée de vie du réseau [43].

Dans ce protocole, quand un nœud capteur est sur le point d'épuiser son énergie, il essaie de trouver un chemin alternatif pour établir une nouvelle connexion avec ses nœuds voisins. Ce chemin alternatif augmente la fiabilité de transmission de données entre les nœuds source et leurs voisins dans la direction de la station de base qui relai les paquets envoyés par ces derniers.

Ce protocole s'exécute en trois phases :

- **Mise en œuvre et établissement de chemin :** Chaque nœud est caractérisé par son identifiant$N_j$, son niveau$HC_j$, son nœud parent $P_j$ et un tableau$A_j$, pour stocker les paquets de données jusqu'à ce qu'un accusé de réception soit reçu. La station de base est initialisée avec HC = 0, P = BS, tandis que les nœuds ordinaires avec d'autres $HC_j = \infty$, $P_j = 1$. Une fois les nœuds déployés, la station de base diffuse un message d'avertissement ADVT ($N_j$, $HC_j$) pour découvrir ses nœuds voisins. Ces nœuds sont considérés comme des nœuds de niveau 1 puisqu'ils se trouvent à un saut de la station de base qui est considérée comme un nœud parent pour ces nœuds de niveau 1. Lorsqu'un nœud reçoit un message ADVT, son HC sera augmenté de un que de celui qui lui a envoyé le message ADVT et il est considéré comme un nœud de niveau N +1 si le nombre de sauts reçu est N. Ainsi, le message ADVT est utilisé pour hiérarchiser le réseau en des niveaux relativement à la station de base.

- **Transmission de données :** Une fois que la hiérarchisation de niveaux est établie, la phase de transmission de données commence. De ce fait, lorsqu'un événement survient au niveau du nœud source. Ce dernier transmet le paquet de données relatif à l'événement au nœud parent et stocke une copie de ce paquet de données dans le tableau. Quand un parent

reçoit le paquet de données émis, il envoie un accusé de réception (ACK) au nœud qui a transmis le paquet. De son côté le nœud source, une fois qu'il reçoit le paquet ACK, il supprime la copie du paquet de données correspondant. Cela continue jusqu'à ce que la station de base reçoit le paquet de données. Un numéro de séquence est attribué à chaque paquet de données transmis pour assurer la fiabilité et garantir sa livraison à la station de base. Si un paquet de données est perdu, il pourra être récupéré à partir du dernier expéditeur.

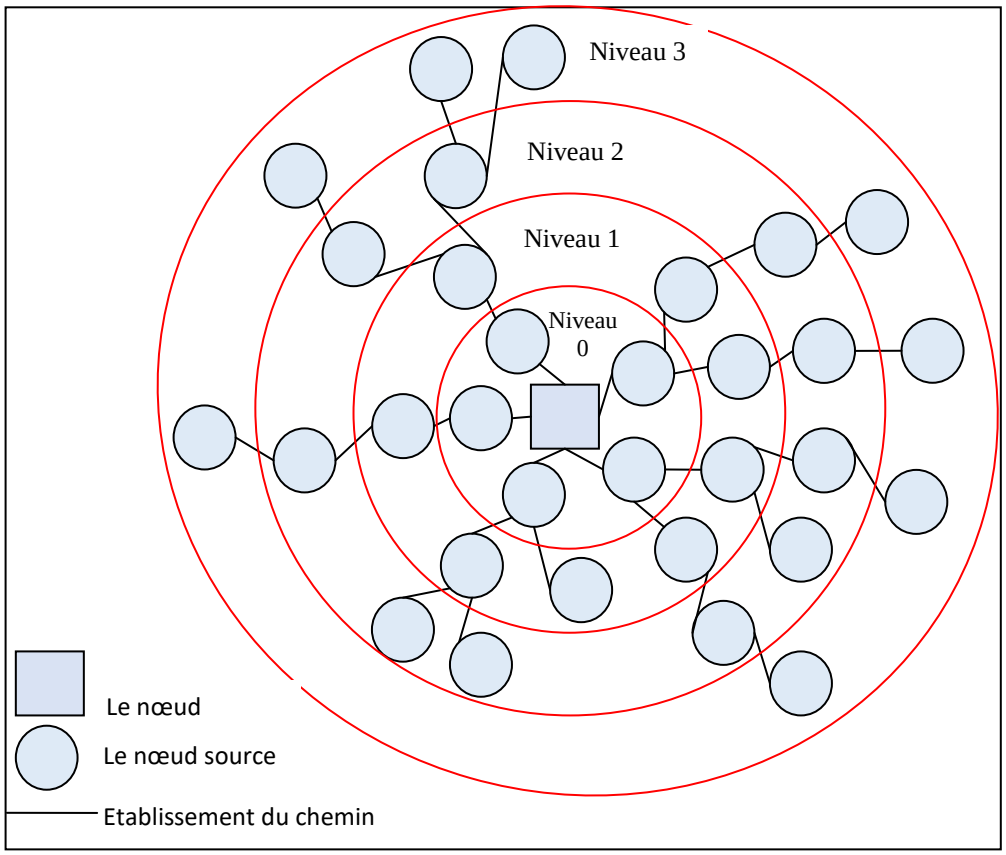



**Figure 3. 2 :** Organisation des nœuds selon ce protocole.

- **Reconstruction du chemin :** Si un nœud est sur le point d'épuiser son énergie, il envoie un message de notification à ses voisins fils en leur demandant de changer leurs nœuds parents pour maintenir la connectivité. Les nœuds fils qui reçoivent ce message, utilisent des paquets « Hello ! » pour découvrir les nouveaux parents ou leurs voisins. Les nœuds fils modifient leurs paramètres ($N_j$, $HC_j$) en fonction de la réponse au message « Hello ! ». Si la réponse provient d'un nœud de niveau inférieur, les nœuds fils gardent leur $N_j$, sinon c'est-à-dire si le message provient d'un nœud voisin, les nœuds fils doivent incrémenter leurs niveaux de 1.

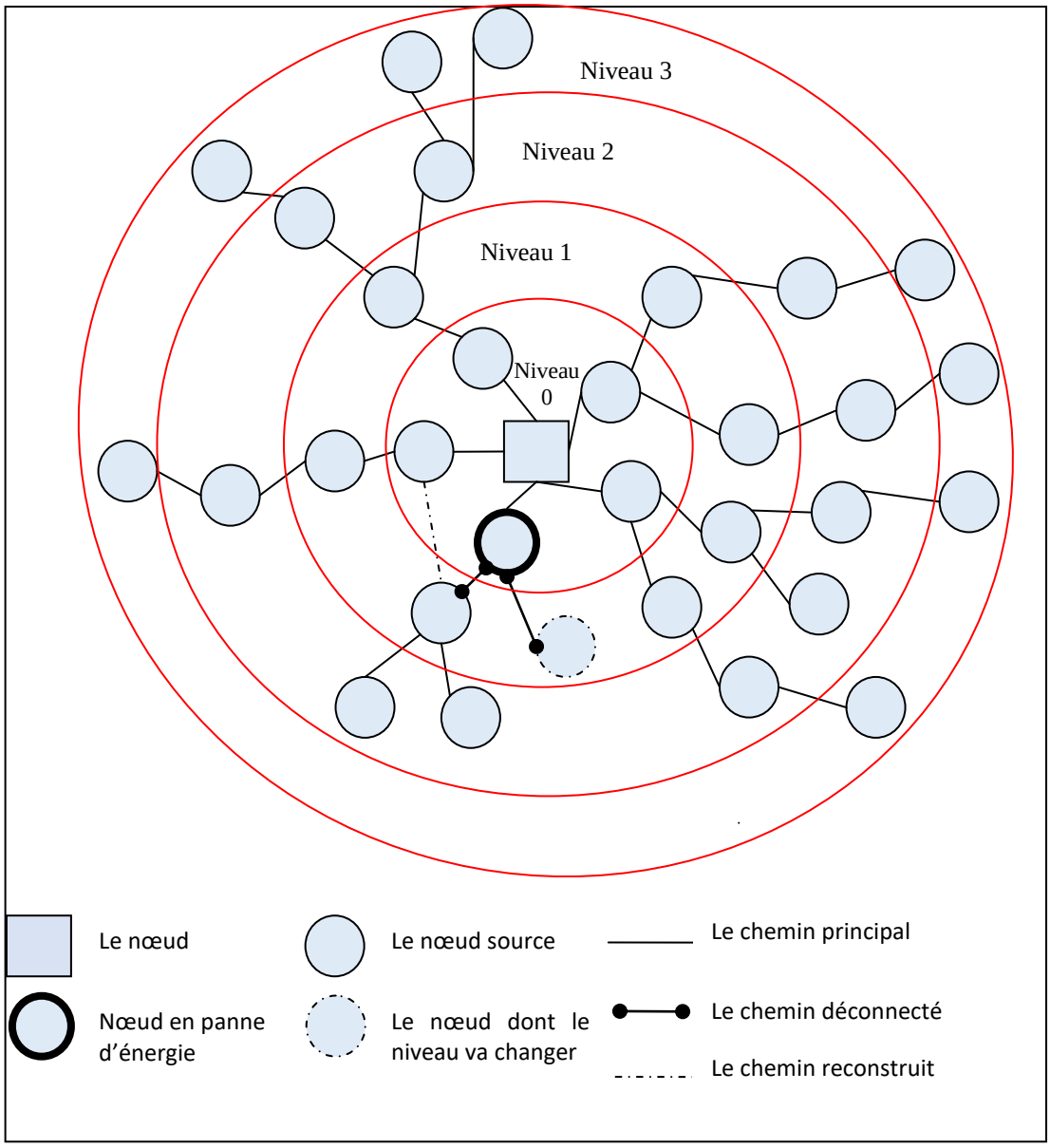



**Figure 3. 3:** Reconstruction du chemin quand le nœud parent épuise son énergie.

La limitation de ce protocole est que le temps pris pour trouver un nouveau nœud voisin affecte la durée de livraison des données. En outre, dans ce protocole, les auteurs ont supposé que l'environnement est idéal car ils n'ont pas pris en compte la présence d'interférences et de bruit qui se trouvent dans le monde réel.

## 3.3.8 Protocole de routage tolérant aux pannes multi-niveaux (FMS)

Le protocole FMS [44] permet de maintenir la connectivité du réseau, même si un nœud est sur le point d'épuiser son énergie. Il permet aussi d'assurer la fiabilité et la rapidité de livraison des données à la station de base car il est conçu pour les applications orientées événement. Généralement, les capteurs sont déployés aléatoirement et en grand nombre. De ce fait, il y aura une redondance dans la livraison de données ce qui a une conséquence sur la durée de vie du réseau de capteurs. Pour remédier à cette limite, FMS permet un ordonnancement d'activité des capteurs en passant un certain nombre de capteurs en mode « veille » sans affecter la fiabilité de livraison de données. Ceci est dans le but d'économiser l'énergie.

Dans FMS, on suppose que chaque nœud possède un identifiant unique, dénoté (Nr), et la communication entre les nœuds voisins est bidirectionnelle. En outre, on suppose que les nœuds sont contraints en termes de puissance de traitement, de stockage et de l'énergie, tandis que la station de base est considérée comme un nœud qui a plus de ressources pour effectuer des tâches ou de communiquer avec les autres nœuds. Le protocole FMS effectue deux opérations de base :

- **Détermination des niveaux des nœuds et établissement de chemin :** cette phase est analogue à celle du protocole cité ci-dessus.

- **Ordonnancement d'activité des capteurs et transmission de données :** L'ordonnancement d'activité des capteurs consiste à faire passer un certain nombre de nœuds périodiquement en mode veille. Au cours de cette période, les nœuds actifs transmettent les paquets de données. Avant qu'un nœud passe en mode veille, il devra informer ses nœuds fils afin qu'ils choisissent un autre nœud parent pour relayer les données. En outre, quand un nœud est en mode veille, il passera en mode actif que si son énergie est supérieure à une certaine valeur seuil. Le choix des nœuds actifs se fait aléatoirement et d'une manière périodique pour que le nœud n'épuise pas son énergie rapidement. Quand un nœud est en mode actif, il participe à l'opération de transmission de données à la station de base. De ce fait, la connectivité est toujours maintenue même si un nœud est mis en mode veille ou il est sur le point de perdre son énergie. Ainsi, FMS est considéré comme un protocole fiable et tolérant aux pannes.

FMS présente les mêmes limitations que le protocole cité précédemment. Il est performant dans un environnement optimal mais ses performances se dégradent dans un environnement réel.

## 3.3.9 Dynamical Jumping Real-Time Fault-Tolerant Routing Protocol for Wireless Sensor Networks (DMRF)

DMRF [45] fonctionne en deux modes de transmission des données : saut-à -saut et "Jumping". Chaque nœud utilise le temps restant pour transmettre un paquet à la station de base et l'ensemble des

nœuds de transfert FCS (Set candidat Forwarding) pour choisir dynamiquement le prochain saut. Quand un nœud présente une défaillance, alors la congestion du réseau ou une région vide se produit. Le mode de transmission sera passé en mode "Jumping", ce qui peut réduire le délai de transmission, et assure la fiabilité de la livraison des paquets de données envoyés à la station de base dans un délai spécifié. Dans DMRF, le processus de transmission est divisé en cinq étapes :

- **Phase d'initialisation :** Dans cette phase, DMRF initialise la liste de voisinage des nœuds, la liste de l'état du réseau (information sur la congestion d'un nœud, les zones vides, …), la liste des candidats FCS, la table des probabilités de transition, et la voie de transmission initiale.

- **Phase de transmission des données :** Dans cette phase, DMRF détecte la défaillance d'un nœud, la congestion du réseau ou une région vide. Le temps restant pour acheminer un paquet de données jusqu'à la station de base sera contrôlé. A partir de ce temps, le paquet sera transmis en mode "Jumping" ou non. Si aucune des conditions ci-dessus ne s'est produite, DMRF sélectionne dynamiquement un membre du FCS comme nœud relais. En outre, une fois les nœuds défaillants sont détectés, ou le temps restant est inférieur à un certain seuil, le mode de transmission "Jumping" sera utilisé.

- **Phase de transmission "Jumping" :** au cours de cette phase, chaque nœud ajuste dynamiquement le contenu de FCS et calcule la probabilité pour transiter par chacun de ces nœuds. Dans ce mode, le paquet de données peut utiliser un saut d'une grande portée pour éviter les nœuds défaillants. Cependant, il ne peut pas garantir le succès de la transmission. Donc, la phase d'ajustement des probabilités de transitions est effectué après chaque transmission "Jumping".

- **La phase d'ajustement des probabilités :** Dans cette phase, DMRF ajuste la probabilité de saut en fonction du résultat de la transmission "Jumping" (succès ou l'échec) et renvoie l'information à son nœud en amont. Lorsque le paquet de données arrive au nœud récepteur, on considère que la transmission est terminée.

DMRF présente certaines limitations en particulier dans le mode "Jumping" qui ne garantit la fiabilité de livraison de données et qui consomme plus d'énergie quand il utilise une grande portée.

### 3.3.10 Algorithme PEQ (Periodic, Event-driven, Query-based):

PEQ [46] combine la conservation d'énergie avec le routage multi-chemins en sélectionnant parmi tous les chemins disponibles, ceux qui consomment moins d'énergie et le considère comme étant chemin principal et les autres des chemins secondaires. En plus de ce mécanisme préventif qui permet un routage fiable avant l'occurrence des pannes, un mécanisme de recouvrement de pannes est

implémenté. Ce dernier remplace le chemin défaillant par un autre chemin qui a des liens fiables et consomme moins d'énergie.

PEQ introduit le paradigme Publish/Subscribe comme montre la figure 3.4 pour l'interaction entre le collecteur et les capteurs simples. En effet, les capteurs envoient des notifications d'événements au collecteur, qui va souscrire son intérêt pour certaines de ces informations. Les capteurs concernés publient par la suite l'information désirée.

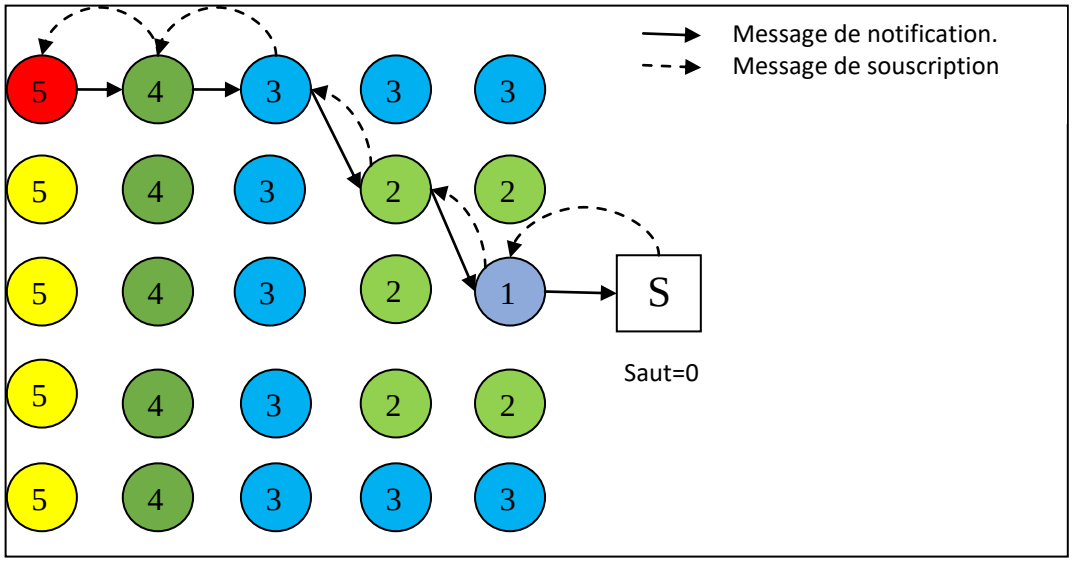
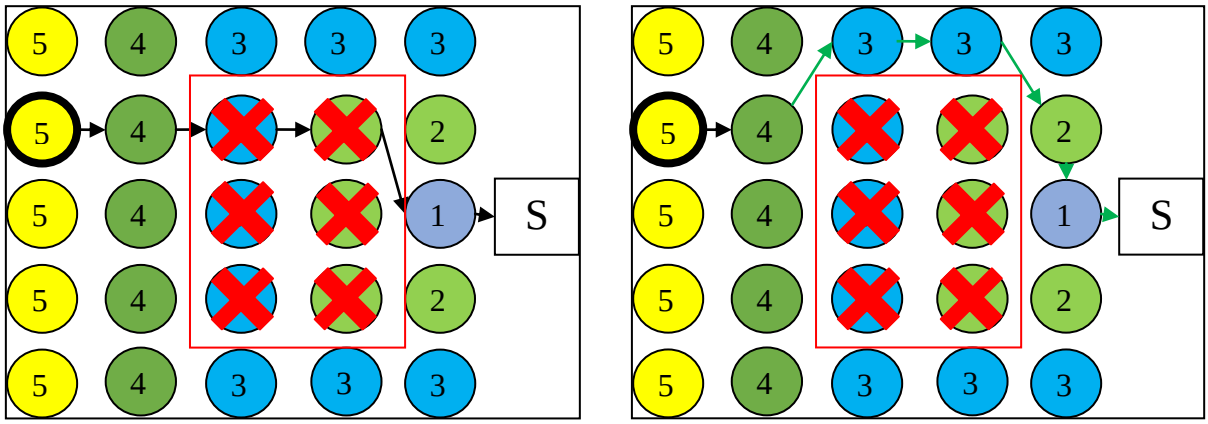



**Figure 3. 4 :** Mécanisme Publication/Souscription

PEQ s'exécute comme suit :

- **Construction de l'arbre de routage :** cet arbre permet de définir les différents chemins multi-sauts possibles pour acheminer les données de chaque nœud à la station de base. Le collecteur commence le processus en initialisant la variable « saut » à 0; par la suite, chaque capteur prend la valeur du saut actuel, l'incrémente puis l'envoie à tous ses voisins. Ainsi la valeur au niveau de chaque capteur désigne le nombre nécessaire de sauts pour communiquer avec le collecteur. A la fin de cette phase seulement les meilleurs chemins sont enregistrés.
- **Transmission de paquets de notification** : chaque capteur envoie selon sa table de routage et l'événement capté, une notification de l'information qu'il a à sa disposition. Pour cela, il utilise le chemin le plus court et le moins coûteux en terme d'énergie.
- **Propagation des paquets de souscription** : dans cette étape, après une souscription, par le collecteur, des données à transmettre, chaque capteur achemine cette dernière jusqu'au capteur concerné.
- **Mécanisme de recouvrement de route :** le recouvrement est effectué après détection de pannes. Un capteur envoie son paquet puis attend un acquittement ACK. S'il le reçoit, le message a été bien transmis ; sinon une panne est détectée au niveau du chemin de routage. On effectue donc une recherche "SEARCH" pour la sélection d'un autre capteur destination tout en minimisant le coût du nouveau chemin. Si aucun capteur n'est trouvé le

capteur devient isolé et doit donc augmenter son rayon de transmission radio pour atteindre les capteurs voisins lointains.




Région avec des nœuds détruits Recouvrement de la panne

**Figure 3. 5:** Recouvrement de routes dans PEQ

## 3.3.11 VTRP (Variable Transmission Range Protocol)

VTRP [47] est une solution d'ajustement du rayon de transmission pour une meilleure propagation de données. Il permet de remédier au problème d'obstacles en les évitant par l'augmentation du rayon de transmission. Ce dernier augmente la probabilité d'atteindre des capteurs actifs quand le rayon actuel utilisé ne couvre aucun capteur à cause de pannes ou d'inactivité des capteurs voisins ou encore dans le cas des réseaux à faible densité. En outre, VTRP offre une meilleure longévité du réseau en évitant l'utilisation fréquente des capteurs critiques (les voisins proches du collecteur) ceci permet d'alléger leur fonction de routage ; conserve leurs batteries et augmente ainsi la durée de vie de tout le réseau. VTRP s'exécute comme suit :

- **Phase de recherche :** Soient p1 et p2 deux capteurs du réseau. Dans la phase de recherche, p2 utilise une diffusion périodique de message afin de découvrir le nœud p1 le plus proche du collecteur. Cependant, une détection de panne est possible si aucun nœud p1 n'est trouvé. Cet échec est causé par l'une des raisons suivantes : soit le nœud p1 est mis en veille ; soit il est en panne ou bien à cause d'un obstacle qui empêche la communication entre p1 et p2.
- **Phase de transmission directe :** En cas où la phase de recherche se termine avec succès, le capteur p2 envoie l'information au capteur p1.
- **Phase d'ajustement du rayon de transmission :** Si la phase de recherche échoue (aucun nœud p n'est détecté) p2 passe à la phase d'ajustement de son rayon de transmission qui représente le cas de recouvrement après pannes. En effet, chaque capteur maintient un compteur local initialisé à 0. A chaque échec de l'étape de recherche, le compteur est incrémenté, et le rayon de transmission R est modifié. Quatre différentes fonctions sont définies selon la vitesse de variation du rayon de transmission : linéaire, multiplicative, exponentielle et aléatoire :

- **Progrès constant :** VTRP est convenable dans ce cas des réseaux où un grand nombre de capteurs est compromis ;
- **Progrès multiplicatif :** $\boldsymbol{VTRP_m}$définit un rayon de transmission qui est augmenté d'une manière radicale. Ce changement offre une meilleure probabilité pour trouver des capteurs actifs. En revanche, il requiert une consommation d'énergie plus importante.
- **Progrès exponentiel :** $\boldsymbol{VTRP_p}$ est une variante qui augmente le rayon d'une vitesse encore plus rapide ;
- **Progrès aléatoire :** quand la densité du réseau n'est pas connue au préalable, on utilise l'approche aléatoire $\boldsymbol{VTRP_r}$pour éviter un mauvais comportement du réseau suite à un mauvais choix.

## 3.3.12 FTEAM (Fault Tolerant and Energy Aware Mechanism)

FTEAM [48], est un mécanisme de tolérance aux pannes qui peut être appliqué aux protocoles de routage basés sur le clustering. FTEAM se déroule en 4 phases: CFS (Cluster Formation State), EFS (Error Free State), CHFS (Cluster Head Failure Rate) et ERS (Error Recovery State). Il définit 3 types de nœuds : les CHs (Cluster Head), les NN's (Nœud Normal) et les SN's (Sleep Nodes).

Lors de la première étape nommé CFS, le CH diffuse un message « *MemberShip* » à tous les NN's . une fois reçu chaque NN calcule son emplacement en utilisant un algorithme RSS interne *(Radio Signal Strength module)*, ce dernier utilise la force du signal des trois autres nœuds pour calculer l'emplacement du nœud cible. Chaque NN sélectionne ensuite le CH approprié et lui envoi un message *« Join »* contenant son ID, sa localisation ainsi que son degré d'énergie. Le CH utilise ensuite ces informations pour calculer la distance entre les nœuds et détecter les nœuds superposés qui détectent des données similaires. Une fois détecté, FTEAM envoie un message *« Sleep »* aux nœuds les plus puissants pour les mettre en veille afin d'économiser leur énergie. Ces derniers seront utilisés pour remplacer les CHs en cas de panne. Ensuite, le CH alloue à chaque nœud de son cluster un slot TDMA ou il pourra envoyer ses données.

Lors de la 2$^{ème}$ phase, tous les nœuds vivants continuent de capter les données et de les envoyer aux CHs qui les agrège et les communique à la SB. Lors de cette phase certains nœuds peuvent tomber en panne d'énergie, cet état continue jusqu'à ce que le degré d'énergie du CH diminue au dessous d'un seuil prédéfini. Lorsque c'est le cas, le cluster passera aux deux autres phases (CHFS et ERS).

En effet lors de ces deux phases les CHs envoient un message « Active » à tous les membres de son cluster, une fois reçu tous les NN s'activent et envoient leur degré d'énergie aux CHs, ces derniers sélectionnent 5% de ces nœuds comme nouveaux CHs. Les nouveaux CHs envoient un message « *Membership* » aux NN de leurs clusters et le processus se répète jusqu'à ce que le réseau échoue.

### 3.3.14 EEBFTC (Extended Energy Balanced Clustering with Fault Tolerance Capability)

EEBFTC [49], est un protocole de routage tolérant aux pannes qui se déroule en deux phases :

- **La phase de configuration :** lors de cette phase la SB détermine les organisateurs, qui forment les clusters et choisissent le noeud le plus puissant en tant que CH.

- **La phase de transmission :** dans laquelle les données sont envoyées au CH, qui se charge de les aggréger et les envoyer à la SB. Lorsque le CH échoue, l'organisateur continuera le rôle de CH pour le temps restant du Round. Ensuite, il choisit le nouveau CH en fonction du niveau de puissance. Si l'organisateur échoue, le CH jouera lui-même le rôle de l'organisateur pour le temps restant du round puis la SB sélectionnera un nouvel organisateur.

### 3.3.15 DFCR (Distributed Fault-tolerant Clustering and Routing)

Dans DFCR [50], la station de base diffuse un message *"HELLO"* à tous les noeuds, et en fonction du RSSI (Received Signal Strength Indication) du message reçu, chaque CH calcule la distance qui le sépare de la station de base. Ensuite, chaque CH diffuse un « hop-packet » indiquant le nombre de sauts pour atteindre la station de base.

De plus, lors du processus de formation des clusters, chaque noeud capteur sélectionne son CH en fonction de l'énergie résiduelle du CH, la distance entre le noeud et le CH et la distance du CH à la SB. Si le nœud n'a pas de CH dans sa plage de communication en raison d'un déploiement aléatoire ou d'une défaillance soudaine du CH correspondant, il diffuse un message *"HELP"* pour rejoindre le CH via un nœud de secours avec une communication multi-hop. En outre, le CH sélectionne le prochain saut pour atteindre la SB selon la distance entre la station de base et le nombre de sauts calculé pendant la phase de configuration

## 3.4 Conclusion

Le routage dans les réseaux de capteurs est un problème complexe car nous devons assurer la fiabilité de livraison de données tout en consommant moins d'énergie. En outre, les protocoles de routage conçus pour les réseaux AD-HOC ne sont pas recommandés pour les RCSF car ces derniers sont composés de nœuds à ressources limitées.

Dans ce chapitre, nous avons présenté les protocoles de routage tolérants aux pannes. Notre constat nous a permis d'illustrer les limites de ces protocoles. D'où, nous avons pensé à améliorer un protocole élaboré au sein de notre laboratoire à savoir le protocole HEEP [65] qui est un protocole meilleur que LEACH et effiace en termes d'énergie mais qui n'est pas tolérant aux pannes.

CHAPITRE 4

# LA CONGESTION DANS LES RESEAUX DE CAPTEURS SANS FIL

## SOMMAIRE

# 4.1 Introduction

Le modèle de communication en amont dans les RCSF est de type many-to-one (nœuds sources vers le puits). En raison de la nature convergente de ce trafic, ce modèle de communication crée un trafic intense dans la région proche du puits que dans le reste de la zone de captage. Ainsi, les nœuds proches du puits devront effectuer plus de tâches que les autres. Etant limités en ressources, ces nœuds se trouvent dans l'état où ils ne peuvent plus effectuer toutes ces tâches. Le terme utilisé pour désigner cette situation est la congestion.

Dans les réseaux sans fil, particulièrement dans les RCSF, il existe plusieurs sources de congestion [51] ; comme le débordement des mémoires tampon, les transmissions concurrentes, la collision des paquets. Parmi les problèmes causés par la congestion, on peut citer : la perte de paquets qui contiennent parfois les informations critiques, le gaspillage de la bande passante, …etc. Il est facile de constater que la congestion dégrade les performances du réseau en l'empêchant de garantir certaines exigences de Qds comme le temps réel, le routage de bout en bout, la maximisation de la durée de vie du réseau, …etc. 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 [52].

Notre objectif dans ce chapitre est de faire une étude bibliographique sur les protocoles ou techniques de contrôle de la 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 présentons les différents types de congestion. Ensuite nous présentons quelques stratégies et protocoles pour le contrôle de la congestion avant la conclusion de ce chapitre.

# 4.2 Les types de congestion

La congestion est provoquée par les sources excédant la capacité (de stockage ou de traitement) des éléments du réseau. Pour cela, deux types de congestion pourraient se produire dans un RCSF [52] :

- **La congestion au niveau du nœud (node-level congestion) :** 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 RCSF, 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 événement critique (incendie dans la forêt, séisme, etc.).
- **La congestion au niveau du lien (link-level congestion) :** elle se produit quand les nœuds capteurs essaient de transmettre en même temps suite à la détection d'un événement critique (incendie dans la forêt, séisme, etc.). Cette congestion conduit à un gaspillage de la bande passante et à un taux d'erreur élevé des paquets lors de leur réception.

# 4.3 Etapes d'un mécanisme de contrôle de la congestion :

- Détection de la congestion
- Notification
- Prise de décision

# 4.4 Les stratégies de contrôle de la congestion

Plusieurs mécanismes de contrôle de la congestion ont été testés, les mécanismes les plus utilisés sont : la perte des paquets, la longueur de la file d'attente, la charge du canal, la durée de vie du paquet, le retard de transmission.

## 4.4.1 Le taux de perte des paquets

Les solutions existantes mesurent cette métrique au niveau de l'émetteur et du récepteur. Elle est mesurée au niveau de l'émetteur en se basant sur les acquittements. En outre lorsque le nœud parent n'écoute aucun message du nœud fils cela peut être considéré comme une perte de paquet.

Le temps de réparation est utilisé dans [53] alors que le rapport de perte est utilisé dans [55,56]. Le principal inconvénient de cette métrique est que les pertes peuvent être dues à des erreurs sans fils.par ailleurs, la fiabilité du paquet n'est pas indispensable pour certaines applications tels que les applications manipulant des données techniques.

## 4.4.2 La longueur de la file

Comme chaque nœud à son buffer, sa taille peut être utilisée comme indicateur de la congestion. En effet lorsque la taille du buffer dépasse un seuil prédéfini ceci peut être utilisé comme indicateur de congestion.

## 4.4.3 La charge du canal

Elle mesure l'activité du canal causé par les transmissions sans fil. Par exemple, pour les radios CC2420, la fonction CCA répond par « 1 » si le canal est occupé ou « 0 » si le canal est vide, la fréquence renvoyée par l'échantillonnage de cette fonction reflète le niveau d'occupation du canal de transmission. La charge du canal est le rapport entre les intervalles ou le canal est occupé au temps total.

En cas d'augmentation des collisions et après plusieurs transmissions infructueuses, les paquets sont supprimés. Par conséquent, la diminution du taux d'occupation du buffer à cause de cette suppression peut induire à l'absence de la congestion et ce lorsque seul l'état du buffer est utilisé pour la détection de la congestion. Par conséquent, pour une détection précise de la congestion, une approche hybride

en utilisant la longueur de la file d'attente et la charge du canal comme indication de la congestion est plus approprié [56].

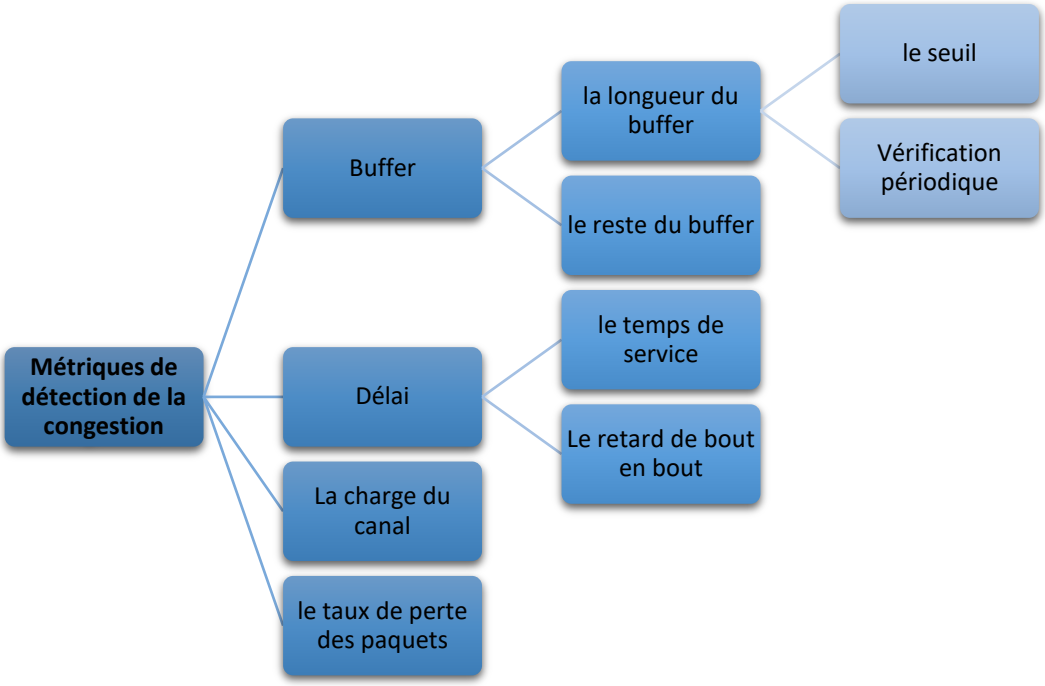



**Figure 4. 1 :** Les métriques de détection de la congestion.

### 4.4.4 Le délai

Il quantifie le temps nécessaire depuis la génération du paquet au niveau de l'émetteur jusqu'à ce qu'il arrive au récepteur, il peut également être calculé comme une partie du retard total.

Le retard d'un seul saut peut être vu comme le temps de service du paquet, qui est le temps séparant l'arrivée des paquets au niveau de la couche MAC et sa transmission avec succès. Il couvre le temps d'attente du paquet, la résolution des collisions et le temps de transmission au niveau de la couche MAC, cette valeur varie en fonction de la longueur de la file d'attente et de la charge du canal.

Une autre mesure du retard est le rapport entre le temps de service du paquet et le temps entre les arrivées des autres paquets.

## 4.5 Classification des protocoles de contrôle de la congestion

Nous pouvons différencier les protocoles de contrôle de congestion à travers plusieurs axes [49] qu'on va décrire dans cette section.

- **Le Mécanisme de détection de la congestion :** Le mécanisme de détection de la congestion peut être local ou global. La détection de congestion locale est réalisée par les nœuds

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

- **L'objectif du contrôle de la congestion :** Dans leur nature, les RCSF sont orientés application. Donc, les protocoles de congestion seront différents selon l'application visée par le RCSF où ils sont appliqués. Pour cette raison, les protocoles de contrôle de congestion sont aussi orientés application.

- **Les mécanismes de contrôle du taux de transfert :** Les mécanismes de contrôle du taux de transfert dans les RCSF peuvent être centralisés, 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.

- **Equité et/ou QoS :** Classiquement, les protocoles de contrôle de la congestion sont chargés de réduire le taux de transmission afin d'éviter ou de réduire la congestion. En plus de cette tâche, d'autres exigences peuvent être envisagées. Cela inclut par exemple, les approches qui essaient de maintenir l'équité des flux opposés 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 [53].

- **Le Modèle de l'application cible :** La plupart des protocoles se basent sur le modèle de communication many-to-one. Cependant, quelques protocoles diffèrent sur leurs hypothèses dans ce modèle, telles que la supposition du flux à haut débit (high-rate flows), ou multiples requêtes à multiples puits (multiple queries to multiple sinks),etc.

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

# 4.6 Quelques protocoles de contrôle de la congestion dans les RCSF

Dans le passé récent, de nouveaux protocoles de contrôle de congestion ont été proposés [53]. Dans le paragraphe suivant nous allons faire une étude un peu détaillée sur certains d'entre eux.

## 4.6.1 Congestion detection and avoidance: CODA

CODA [57] est une technique de contrôle de congestion pour les RCSF qui comprend trois mécanismes :

- **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 du 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.

- **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.

- **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.

## 4.6.2 Event-to-Sink Reliable Transport (ESRT)

Le protocole ESRT [58] 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.

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 envoyer s'il constate que le 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 4.2. 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. Le but d'ESRT est de maintenir toujours l'état du réseau dans OOR.

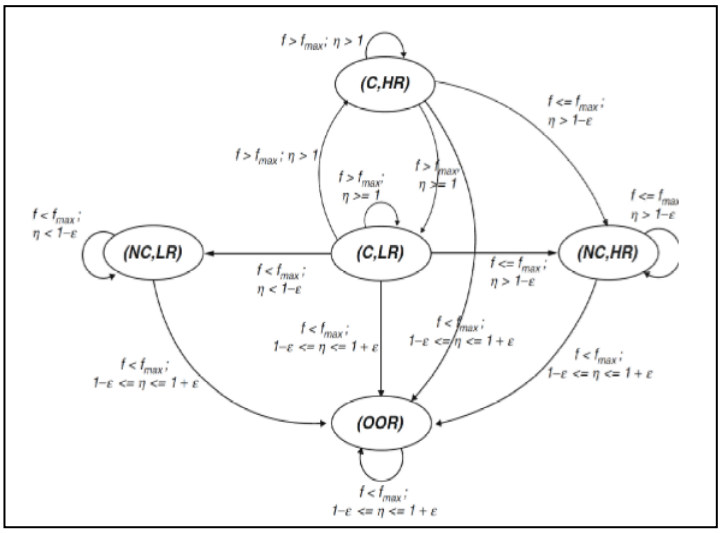



**Figure 4. 2 :** Le diagramme de transition de ESRT.

## 4.6.3 Fair Rate Allocation (FRA)

Fair Rate Allocation (FRA), ou allocation équitable de taux de transmission est une approche explicite au contrôle de congestion avec garantie d'équité proposée par [59]. Le mécanisme de FRA comprend les trois démarches suivantes :

Déterminer 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 ou la couche transport envoie le paquet à la couche réseau jusqu'au moment ou la couche réseau signale que le paquet a été transmis.

Assigner 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 :

$$r_{data} = \frac{r}{n}$$

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 nombres 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 propager 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 4.3. 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.

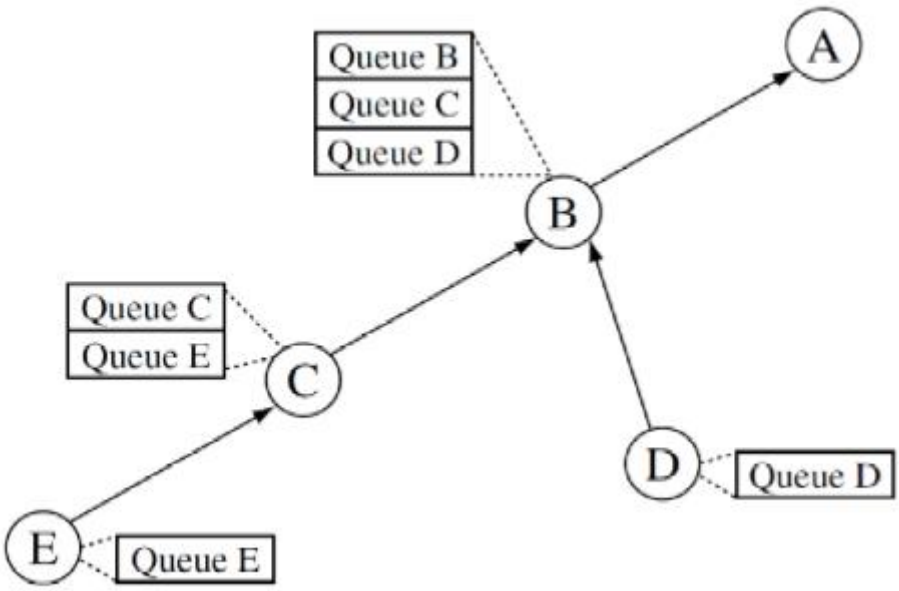



**Figure 4. 3 :** FIFO multiples pour assurer la délivrance équitable de données dans FRA.

## 4.6.4 Autres protocoles

Le nombre élevé des protocoles de contrôle de congestion ne nous donne pas la possiblité 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 vus précédemment. Nous les résumons dans les paragraphes suivants.

Dans [60, 61], les auteurs proposent le protocole IFRC (Interferance-Aware Fair Rate Control) qui prend en considération les interférences pour contrôler la congestion.

Dans [62], 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).

# 4.7Conclusion

Le modèle de communication « many to one » dans les RCSF, le partage du canal de transmission ainsi que la limite des ressources dont sont équipés les nœuds capteurs sont les principaux éléments que peuvent conduire à un réseau congestionné. La congestion lorsqu'elle existe dans un réseau, elle dégrade ses performances et peut causer beaucoup de problèmes.

Dans ce chapitre nous avons identifié quelques problèmes parmi lesquels nous citons : la perte des paquets contenant des informations parfois critiques, le gaspillage de la bande passante, …etc. Nous avons également présenté quelques protocoles proposés pour détecter et/ou contrôler la congestion dans les RCSF. Les notions présentées dans ce chapitre sont nécessaires pour la compréhension de la solution que nous avons proposée dans le chapitre 5.

CHAPITRE 5

# FT-HEEP : PROTOCOLE CROSS LAYER TOLERANT AUX PANNES

## SOMMAIRE

# 5.1 Introduction

Dans les chapitres précédents, nous avons présenté un état de l'art sur les différents protocoles de routage tolérants aux pannes qui ont été proposés pour les RCSF. Chacun de ces derniers à ses avantages, ses inconvénients ainsi que des applications pour lesquelles il a été développé, c'est la cause pour laquelle la recherche dans cet axe ainsi que dans les autres axes de recherche dans les réseaux de capteurs sans fil reste ouverte à de nouvelles idées afin d'optimiser les protocoles existants ou même pour proposer de nouveaux protocoles qui optimisent les performances du réseau.

Dans ce chapitre nous allons présenter notre nouveau protocole de routage qui assure un routage des données même en cas de panne de certains nœuds et ou même du CH. Ce dernier est nommé FT-HEEP [66] (Fault Tolerant Hybrid Energy Efficient Protocole) qui est une extension du protocole HEEP (Hybrid Energy Efficient Protocol) [65]. Contrairement au protocole HEEP d'origine qui opère au niveau de la couche réseau et qui ne fourni aucun mécanisme qui permet d'acheminer les données en cas de panne de certains nœuds ou du CH, notre nouveau protocole est un protocole Cross-layer qui permet non seulement d'acheminer les données même en cas de panne mais aussi de choisir le nœud ayant la meilleure qualité du lien et le moins congestionné pour le faire.

Ce chapitre est organisé comme suit : la première section est réservée à une présentation du concept d'architecture Cross-layer, la section 2 est réservée à une présentation du protocole HEEP, la section 3 est réservée à la présentation de notre nouveau protocole. Quand à la section 4, elle discute les résultats d'expérimentation.

# 5.2 Le concept Cross-Layer

Se réfère à la conception de protocole en exploitant activement les dépendances entre les couches (adjacentes ou non) afin d'obtenir des gains de performance. Ces couches peuvent être adjacentes, ou même non adjacentes dans la pile protocolaire [63].

La flexibilité de ce concept aide à améliorer les performances de communication de bout en bout. Cependant, l'approche Cross-Layer peut augmenter de manière significative la complexité de conception

# 5.2.1 Les communications Cross-Layer

On distingue quatre classes de communication Cross-Layer:

- **Flux d'informations de haut en bas (downward)** : Les couches supérieures fournissent des paramètres de configuration aux couches inférieures.
- **Flux d'informations de bas en haut (upward) :** Les couches supérieures requièrent des informations provenant des couches inférieures.
- **Flux d'informations bidirectionnel :** deux couches différentes peuvent collaborer entre elles en échangeant des informations.

- **Fusion de couches adjacentes :** plusieurs couches adjacentes sont conçues ensemble pour former une « super couche ». Ainsi, le service fourni par cette super couche est la collection des services fournis par ces couches adjacentes.

## 5.2.2 Importance de l'approche inter-couches (Cross-Layer) dans les RCSF

Les exigences rigoureuses des réseaux de capteurs poussent de plus en plus vers la conception des solutions sous une architecture inter-couches. En effet, afin d'atteindre les performances souhaitées sous multiples contraintes (QdS, énergie, distorsion, sécurité, ...), les protocoles et les algorithmes des couches MAC, réseau, transport et application sont amenés à fonctionner d'une manière encore plus interactive. Dans un tel contexte, chaque couche est consciente de l'importance des données en cours de traitement pour l'ensemble du système.

Les approches basées sur le concept du Cross-layer permettent généralement d'obtenir de meilleures performances par rapport aux approches qui respectent la notion d'abstraction des couches définie dans le modèle OSI [64]. Cependant, ces approches sont généralement plus complexes à mettre en ouvre et sont généralement moins compatibles avec les protocoles et normes existants et peuvent être, ainsi difficilement réutilisable. Cela donne naissance à un compromis entre performance et interopérabilité.

## 5.2.3 Les types d'architecture Cross-Layer

Les architectures Cross-layer peuvent être classées en 3 catégories : Architecture Cross-layer à base de communication directe, architecture Cross-layer à base de communication indirecte et architecture Cross-layer à base de nouvelles abstractions [63].

### 5.2.3.1 Architecture Cross-layer à base de communication directe

Cette architecture permet aux couches, même si elles ne sont pas adjacentes, de communiquer directement entre elles (voir la Figure 5.1), afin d'optimiser la QdS par exemple. Pour cela, il est nécessaire de créer des nouvelles interfaces, d'intégrer des nouvelles routines aux couches qui leurs permettront la réception et le traitement des données Cross-Layer. Cependant, il faut noter que le nombre de routines à implémenter sera variable selon le nombre de protocoles à satisfaire. De plus, cette méthode ajoute un certain nombre de contraintes telles que le ralentissement de l'exécution du code puisque le code Cross-Layer a été ajouté, par conséquent une mise à jour difficile à maintenir.

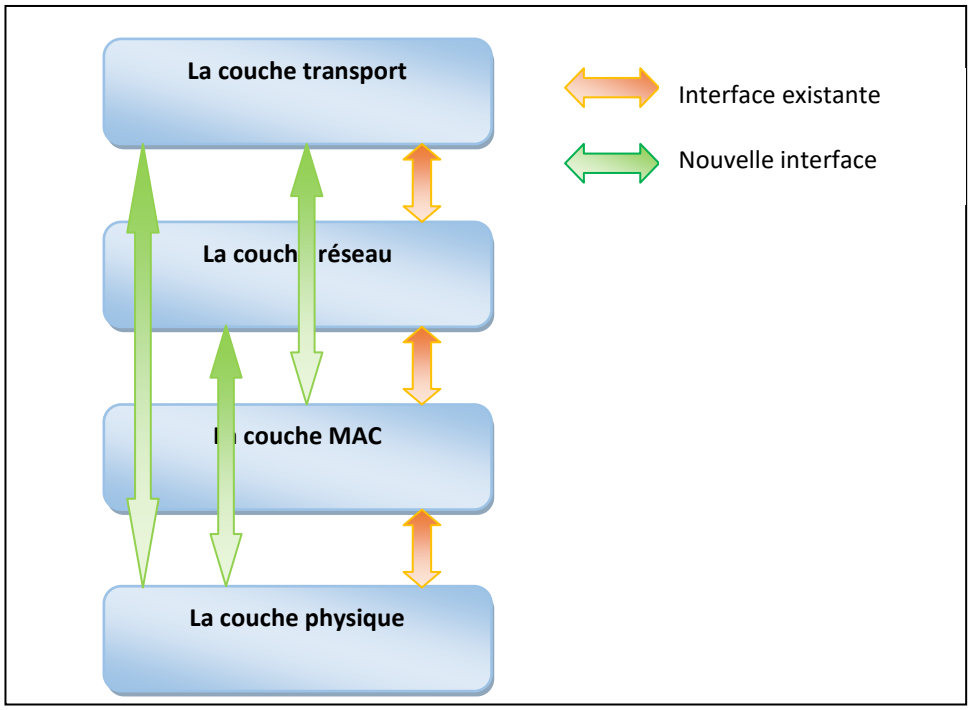



**Figure 5. 1.** Architecture Cross-layer à base de communication directe

### 5.2.3.2 Architecture Cross-layer à base de communication indirecte

Dans cette architecture, une entité intermédiaire se charge des communications entre les différentes couches protocolaires (voir la figure 5.2). Par conséquence, le fonctionnement normal de la pile protocolaire est conservé, ce qui permet de ne pas redéfinir les protocoles existants (architecture en couches). La dénomination et les fonctionnalités de l'entité intermédiaire varient selon l'architecture Cross-layer [159,160]. L'utilisation d'une interface de communication permet de :

- Maintenir l'architecture classique en couches, étant donné qu'il n'y a pas de modification sur ses fonctionnalités de base (compatibilité complète).
- Profiter des avantages de la conception modulaire de l'architecture en couches, ce qui facilite le processus de mise à jour. Ainsi, l'addition et/ou la suppression des protocoles ne va pas engendrer des modifications dans les autres couches.
- Mettre à jour l'entité cross-layer sans avoir à gêner les protocoles relatifs aux couches.

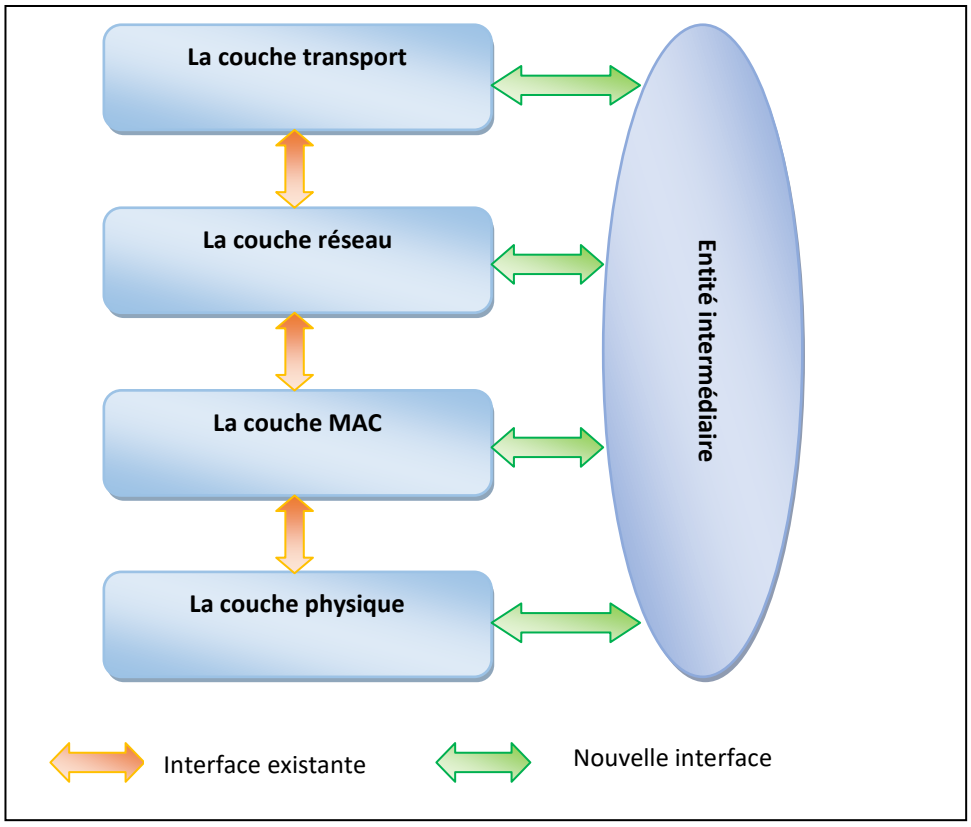



**Figure 5. 2.** Architecture Cross-layer à base de communication indirecte

### 5.2.3.3 Architecture Cross-layer à base de nouvelles abstractions

Une troisième catégorie s'abstrait complètement du modèle en couche (voir la figure 5.3), elle est donc bien plus flexible mais elle viole complètement les préceptes du modèle en couches. Dans cette approche le concept d'architecture en couches est totalement abandonné, puisque plusieurs couches peuvent être couplées ensembles afin de construire une sorte de super couche, qui peut gérer différentes tâches dans le réseau. Cette nouvelle approche Cross-layer permet d'offrir une forte flexibilité avec un minimum de problèmes.

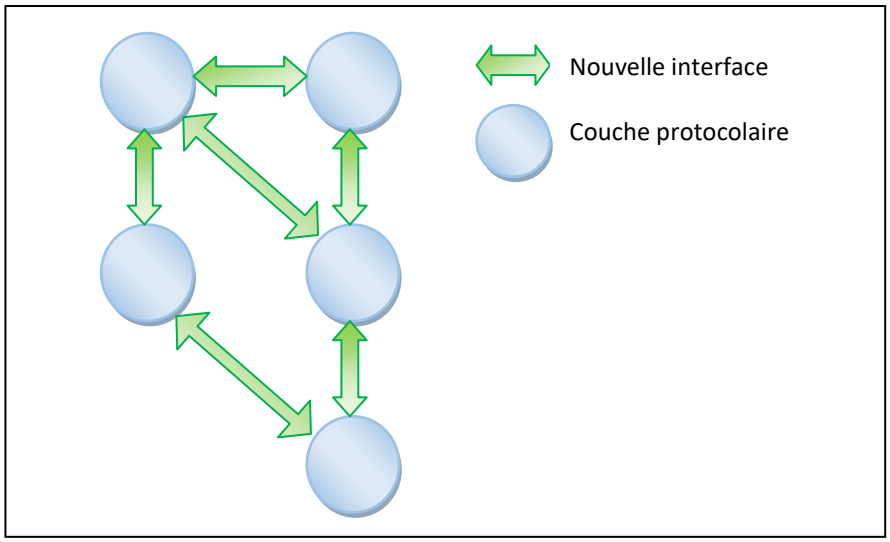



**Figure 5. 3.** Architecture Cross-layer à base de nouvelles abstractions

## 5.3 Les concepts de base du protocole HEEP (Hybrid Energy Efficient Protocol)

Le protocole HEEP [65] est un protocole de routage qui opère au niveau de la couche réseau et qui se base sur les deux protocoles LEACH [12] et PEGASIS [13]. Dans HEEP les nœuds appartenant au même Cluster sont organisés sous forme d'une chaîne ce qui permet d'améliorer la dissipation de l'énergie et par conséquent de réduire la charge sur le CH (cluster-head). En effet, chaque nœud de la chaine ne communique qu'avec son voisin le plus proche et ne peut communiquer directement avec le CH ce qui permet d'économiser de l'énergie et d'offrir une meilleure utilisation de la bande (voir la figure 5.4). De plus, un mécansime d'agrégation des données est introduit au niveau de chaque nœud de la chaîne ce qui réduit la quantité de données échangées entre les nœuds et le CH, et préserve l'énergie des noeuds. En effet, lorsqu'un nœud reçoit un paquet de données, il agrège les données reçues avec les siennes et les envois à son autre voisin jusqu'à atteindre le CH qui les transmet directement à la SB [65].

A l'inverse de LEACH, le nombre de nœuds qui communiquent avec le CH est considérablement réduit. Ceci implique une meilleure économie d'énergie et prolonge le temps de vie des CHs, car si ces derniers meurent (épuisent leur réserve d'énergie), tous les nœuds du cluster vont perdre leur pouvoir de communication avec la SB et par conséquent le cluster tout entier est considéré comme invalide (ne communique pas avec la BS).

Le protocole HEEP adopte le concept de la rotation aléatoire du rôle de CH proposé par LEACH, qui régule la dissipation de l'énergie permettant d'éviter que les nœuds choisis comme CHs meurent plus rapidement. Cependant, par opposition à LEACH, HEEP réutilise le concept de PEGASIS, il organise les nœuds du cluster sous forme de chaîne, ce qui a pour effet de prévenir que les nœuds les plus éloignés des CHs épuisent leurs réserves d'énergie. En ce qui concerne l'accès au medium (Media access), la même méthode proposée dans LEACH a été utilisé, où les CHs établissent un plan de transmission (TDMA schedule) qui attribue à chaque nœud le temps exact pendant lequel il doit transmettre les données collectées [67].

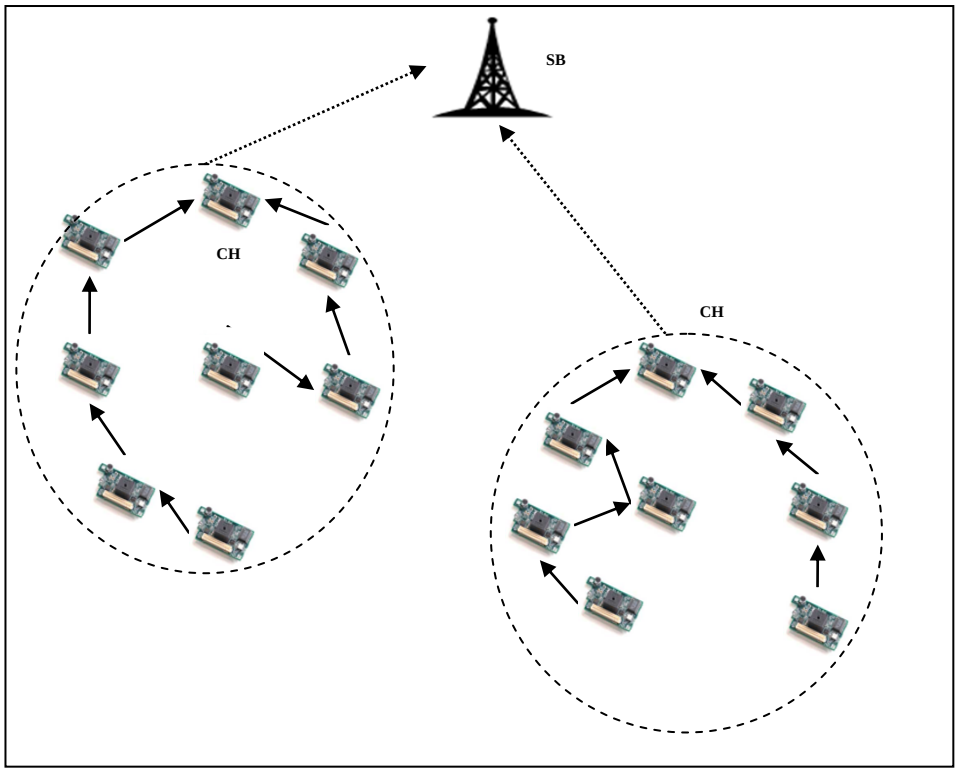



**Figure 5. 4.** Organisation des nœuds selon HEEP.

## 5.4 Les grandes étapes du protocole HEEP

Le déroulement du protocole HEEP est divisé en plusieurs cycles d'exécution. Chaque cycle commence par une phase d'initialisation dans laquelle les clusters à chaînes sont formés et les CHs sont élus, suivie d'une phase de transmission où les données collectées sont transmises à travers les chaînes aux CHs, ces derniers les transmettent à leur tour à la station de base. Les nœuds doivent être synchronisés de façon à participer à la phase d'initialisation en même temps. La Figure 5.5 montre les étapes d'exécution du protocole HEEP. A chaque phase de transmission, tous les nœuds appartenant au même cluster sont délégués pour la tâche de la collecte des données (tâche à faible consommation énergétique), tandis que la tâche la plus coûteuse en énergie (la tâche de transmission des données vers la SB) est assignée au nœud possédant la plus grande réserve d'énergie, ce dernier est le CH. Cela signifie que HEEP délègue la tâche la plus coûteuse en énergie à un seul nœud dans le cluster à chaque phase de transmission et assigne la tâche de collecte aux nœuds restants, même s'il y a plusieurs nœuds puissants dans le cluster. Afin de minimiser les problèmes d'interférence et d'overhead, la durée de la phase d'initialisation est fixée de façon à être beaucoup plus petite par rapport à la phase de transmission.

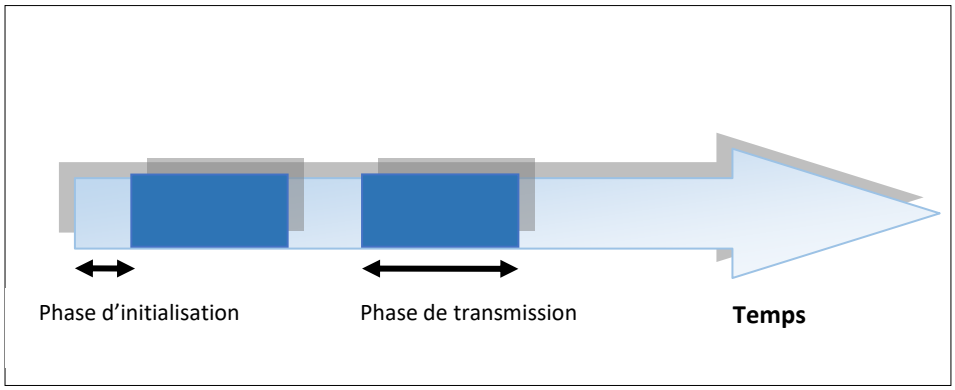



**Figure 5. 5.** Les étapes d'exécution du protocole HEEP.

Le coût énergétique de la phase d'initialisation est très faible comparé à la phase de transmission, et dépend du nombre de nœuds dans le réseau et la distance de la SB par rapport au réseau, nous pouvons l'estimer par la règle suivante :

$$\sum_{i=1}^{i=N} q_i \, E_{elec} + q_i E_{fs} d^2 \quad \text{.....(1)}$$

Où :

$q_i$: est la taille d'un paquet de contrôle transmis par un nœud $i$.

$E_{elec}$ : est l'énergie consommée par les circuits électroniques (énergie de calcul).

$E_{fs}$ : est l'énergie perdue dans l'espace de transmission,

$d$ : est la distance géographique entre un nœud $i$ et la station de base, et $N$ est le nombre de nœuds vivants dans le réseau.

Le total d'énergie initiale dans le réseau est défini par l'équation suivante :

$$E_{Total} = NE_0 \quad \text{.....(2)}$$

$E_0$ Est l'énergie initiale de chaque nœud dans le réseau et $N$ est le nombre de nœuds du réseau.

La durée de vie du réseau est divisée en plusieurs phases de communication (cycle). Nous estimons le nombre de ces dernières par l'équation suivante :

$$NB_{round} = \frac{E_{Total}}{E_{round}} \quad \text{.....(3)}$$

Où :

$$E_{round} = q[2NE_{elec} + NE_{DA} + NE_{fs} d^2{}_{\,to\,next\,node} d^2 + E_{mp} d^4_{CH\,to\,BS}] \quad \text{.....(4)}$$

$E_{DA}$ Est l'énergie consommée durant l'agrégation des données, $d_{CH\,to\,BS}$ est la distance moyenne entre les CHs et la BS, $d_{to\,next\,node}$ est la distance moyenne entre les nœuds voisins dans la chaine, $q$ est la taille des paquets de données à transmettre, $N$ est le nombre de CHs dans le réseau $E_{fs}$ et $E_{mp}$ représentent l'énergie perdue dans l'espace de transmission. Étant donné que les distances de transmission sont réduites, le nombre de phases de communication est impérativement optimisé. Le total des distances de transmission peut être calculé par la formule suivante :

$$d_{Total} = \iint \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} dxdy \quad \text{.....(5)}$$

Où $X_j$ et $Y_j$ sont les coordonnées du prochain nœud dans la chaine de transmission.

# 5.5 Le protocole FT-HEEP (Fault Tolerant Hybrid Energy Efficiency Protocol)

Nous proposons dans ce chapitre un nouveau protocole qui assure une continuité de fonctionnement même en cas de panne. Contrairement au protocole originale et aux autres protocoles mono couche qui assurent un mécanisme de tolérance aux pannes, notre nouveau protocole est un protocole cross layer. Ce dernier est une extension du protocole HEEP appelé FT-HEEP (Fault Tolerant Hybrid Energy Efficiency Protocol) [66]. Contrairement au protocole HEEP qui opère au niveau de la couche réseau, FT-HEEP est un protocole cross layer qui exploite l'interaction entre les couches adjacentes.

L'architecture cross-layer proposée conserve la structure en couches traditionnelle et considère l'interaction directe entre les trois couches : couche physique, MAC et réseau comme le montre la figure ci-dessus :

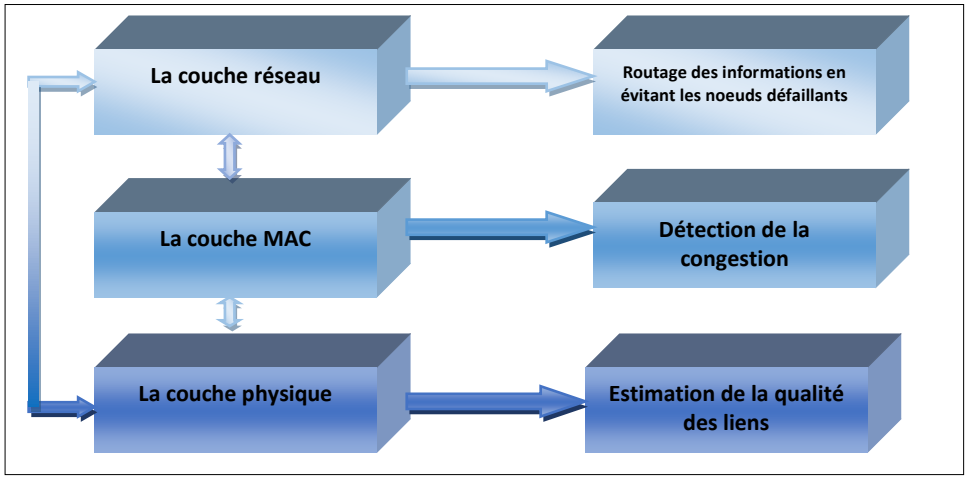



**Figure 5. 6.** L'architecture Cross layer proposée.

Au niveau de la couche physique, une estimation de la qualité des liens est effectuée, cette information est utilisée par la couche réseau pour sélectionner le nœud ayant une meilleure qualité des liens comme nœud secours en cas de panne. Au niveau de la couche MAC, un mécanisme permettant d'éviter les nœuds congestionnés est proposé, cette information est également utilisée par la couche réseau. Et enfin un mécanisme de tolérance aux pannes est proposé au niveau de la couche réseau.

## 5.5.1. Evaluation de la qualité des liens radio au niveau de la couche physique

La couche physique assure l'activation et la désactivation du medium radio, la sélection du canal de communication, le test de l'occupation du canal noté CCA (pour Clear Channel Assessment), la détection d'énergie notée ED (pour Energy Detection) sur un canal, l'indication de la qualité d'un lien noté LQI (Link Quality Indicator), ainsi que la transmission et la réception des bits à travers le canal.

L'estimation de qualité du lien (LQE) [68] est un outil fondamental dans les communications sans fil pour augmenter la fiabilité des liens. Dans ce travail, nous avons sélectionné le taux de réception de paquets (PRR) [69] comme métrique d'estimation de la qualité de lien (LQE) [70]. Le paramètre PRR est

défini comme le rapport du nombre de paquets reçus avec succès sur le nombre total de paquets envoyés.

Cette métrique est calculée au niveau de chaque nœud dans le réseau, ces derniers le font en envoyant en brodcast N paquets d'estimation de la qualité des liens, un laps de temps t est introduit afin d'éviter la collision entre les paquets. Chaque nœud qui reçoit ce paquet transmet un message "*Hello!*". Le nœud source estime sa qualité de liens comme suit :

$$PRR_m = \frac{Nombre\ de\ paquets\ \text{"Hello!" reçus}}{Nombre\ total\ de\ paquets\ émis} \quad \text{.....(6)}$$

$$batt_N = \frac{batt_t - batt_{critical}}{batt_{max}} \quad \text{.....(7)}$$

$$L = batt_N * PRR_m \quad \text{.....(8)}$$

Ou : $PRR_m$ est le taux de réception des paquets, $L$ est l'indicateur de la qualité des liens, $batt_N$ est le niveau d'énergie restant, $batt_t$ est le degré d'énergie restant au moment *t*, $batt_{critical}$ est le niveau d'énergie critique et $batt_{max}$ est le degré maximal d'énergie.

## 5.5.2. Contrôle de la congestion au niveau de la couche MAC

Dans ce travail, nous avons proposés une nouvelle métrique pour le contrôle de la congestion appelée « congestion Ratio (CR) ». Elle est calculée en fonction des trois paramètres les plus importants pour améliorer les performances du réseau et réduire la congestion qui sont : le taux de remplissage du Buffer,le taux de la bande passante et le taux de puissance restante.

Le taux de remplissage du Buffer est calculé comme suit :

$$Le\ taux\ de\ remplissage\ du\ buffer = {La\ taille\ actuellement\ occupée}\ /_{La\ taille\ disponible} \quad \text{.....(9)}$$

Une période de sommeil est introduite pour calculer le taux de la bande passante disponible. Ce dernier peut être calculé comme suit :

$$le\ taux\ BW_{AV} = BW_{MAX}({Idle_t}/_{Int_t}) \quad \text{.....(10)}$$

Ou : $le\ taux\ BW_{MAX}$est le taux maximale de bande passante disponible, $Idle_t$ est la période de sommeil du canal sans fil pendant l'intervalle $Int_t$, qui peut être calculée comme suit :

$$Idle_t = Int_t - Busy_t \quad \text{.....(11)}$$

Ou : $Busy_t$ est la période d'occupation du canal sans fil

Et enfin la métrique CR est calculée à partir des équations (9),(10) et (11) comme suit :

$$CR = {(le\ taux\ de\ remplissage\ du\ buffer + taux\ BW_{AV} + taux\ E_t)}/_3 \quad \text{.....(12)}$$

## 5.5.3. Mécanisme de tolérance aux pannes au niveau de la couche réseau

### 5.5.3.1 Mécanisme de détection de la panne des nœuds et des CHs :

La détection de pannes est la première phase de gestion de pannes, où un échec inattendu devrait être correctement identifié. Dans notre nouveau protocole, nous avons proposé deux approches pour la détection des pannes : une approche préventive et une approche corrective

#### *a. Approche préventive*

Lors de la phase d'initialisation, et après l'envoi de l'ID, la localisation, CR (Congestion Ratio), une estimation de la qualité des liens ainsi que le degré d'énergie de chaque nœud à la SB, cette dernière vérifie le degré d'énergie de chaque nœud, si la réserve d'énergie de ce dernier a atteint un niveau bas (<= seuil), elle le supprime de la liste des nœuds qui vont participer au routage des informations et le supprime également de son plan TDMA ceci permet d'éviter les nœuds ayant un faible degré d'énergie car ils tombent en panne rapidement. De plus, elle vérifie la qualité des liens de ce nœud si sa qualité est inférieure à la qualité moyenne des nœuds du réseau elle l'élimine temporairement de la liste des nœuds qui vont participer au routage des informations et lui attribue un slot TDMA dès que la qualité moyenne des noeuds du réseau devient inférieur ou égale à sa qualité. Ensuite elle vérifie son CR, s'il est supérieur au seuil, elle lui attribue un slot TDMA (voir la figure 5.7).

L'objectif de cette approche préventive est d'éviter :

- Les pannes dues au problème d'énergie, car les nœuds peuvent transmettre des paquets érronés lorsqu'ils n'ont pas assez d'énergie ce qui provoque un dysfonctionnement du réseau.
- L'utilisation de liens de mauvaise qualité afin de s'assurer que seuls les nœuds ayant des liens forts et stables sont choisis pour le routage.
- Les nœuds encombrés ou qui se trouvent dans des zones encombrées, et ce afin d'équilibrer la charge de manière égale sur le réseau.

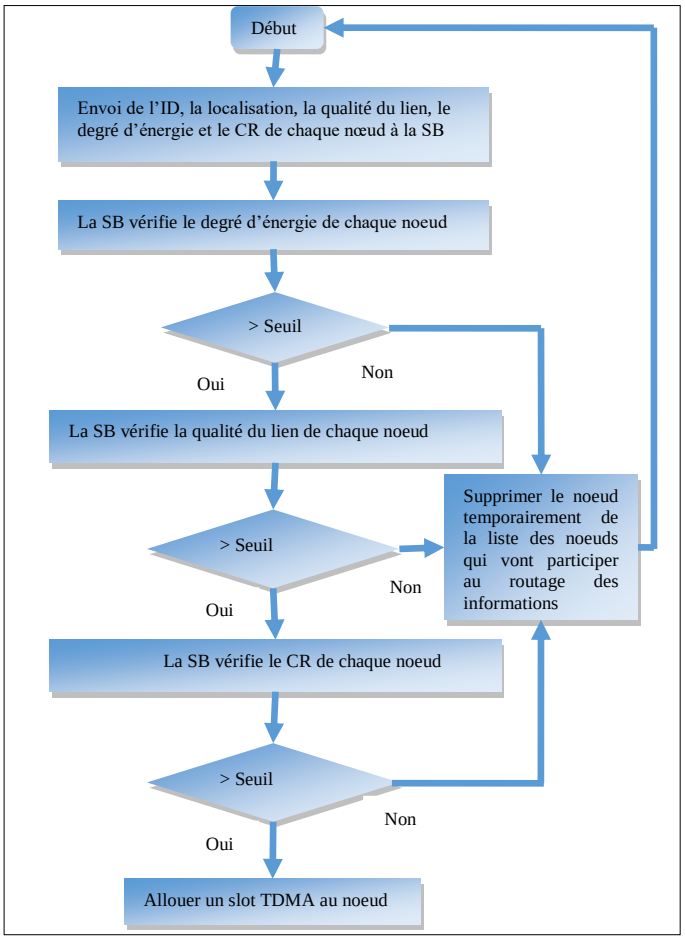



**Figure 5. 7.** Organigramme de l'approche préventive.

## *b. Approche corrective*

Lors de la phase de transmission, chaque nœud y compris le CH doit générer un paquet « Hello ! » pendant son slot TDMA et avant communication des données, ce qui permettra à ses proches voisins de la chaine de connaitre son état.

Si le nœud génère ce paquet cela signifie qu'il est toujours vivant, le cas échéant le nœud qui lui a communiqué les données le déclare comme en panne au CH et passe à la procédure de recouvrement. Si ce nœud est lui-même un CH le dernier nœud de la chaine déclare le CH comme en panne à la SB.

Cela signifie que chaque nœud génère deux types de message : un message *« Hello ! »* pour informer son prédécesseur qu'il n'est pas en panne et un message contenant les données captées. Et reçoit un message contenant les données captées par son prédécesseur comme indiqué dans le code ci-après.

```
set mac_dst $MAC_BROADCAST
set link_dst $sender
set spreading_factor $opt(spreading)
set msg $HELLO
set datasize [expr $spreading_factor * \[expr [expr $BYTES_ID *
[llength $msg]] + $opt(sig_size)]]

$self send $mac_dst $link_dst $HELLO $msg $datasize 1000 $code_
puts "Send source BS: message de hello destination:$sender
(time:[$ns_ now])"
```

**Figure 5. 8.** Code d'implémentation des messages « Hello ! »

Pour réaliser ce mécanisme on doit prolonger la période de temps alloué à chaque nœud du réseau comme suit :

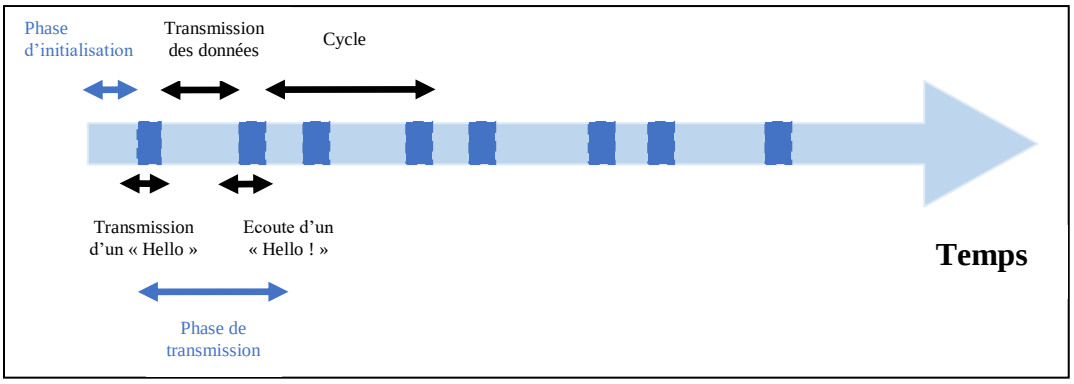



**Figure 5. 9.** Etapes d'exécution de notre protocole.

### 5.5.3.2 Mécanisme de recouvrement de la panne des nœuds et des CHs :

Le recouvrement des pannes est la phase dans laquelle on effectue des opérations d'élimination des effets de pannes. Contrairement aux protocoles de tolérance aux pannes existants, qui proposent des mécanismes soit pour le revouvrement des pannes des nœuds soit pour le recouvrement des pannes de CHs. Notre nouveau protocole propose des mécanismes pour le recouvrement des pannes des nœuds et des CHs.

#### *a. Recouvrement de la panne des nœuds :*

Après formation de la chaine, chaque nœud choisi le voisin de son voisin comme nœud de secours, si jamais son voisin tombe en panne le nœud de secours se charge d'acheminer les données à travers la chaine jusqu'à ce qu'elles arrivent au CH c'est-à-dire que lorsque le nœud n'écoute aucun message *« Hello ! »* de son voisin il le déclare comme en panne et renvoie les données vers le nœud de secours comme illustré dans la figure 5.10 :

Par exemple lorsque le nœud 4 tombe en panne, le nœud 5 utilise son nœud de secours qui est le nœud 3 pour acheminer les données à travers la chaine jusqu'à ce qu'elles arrivent au CH.

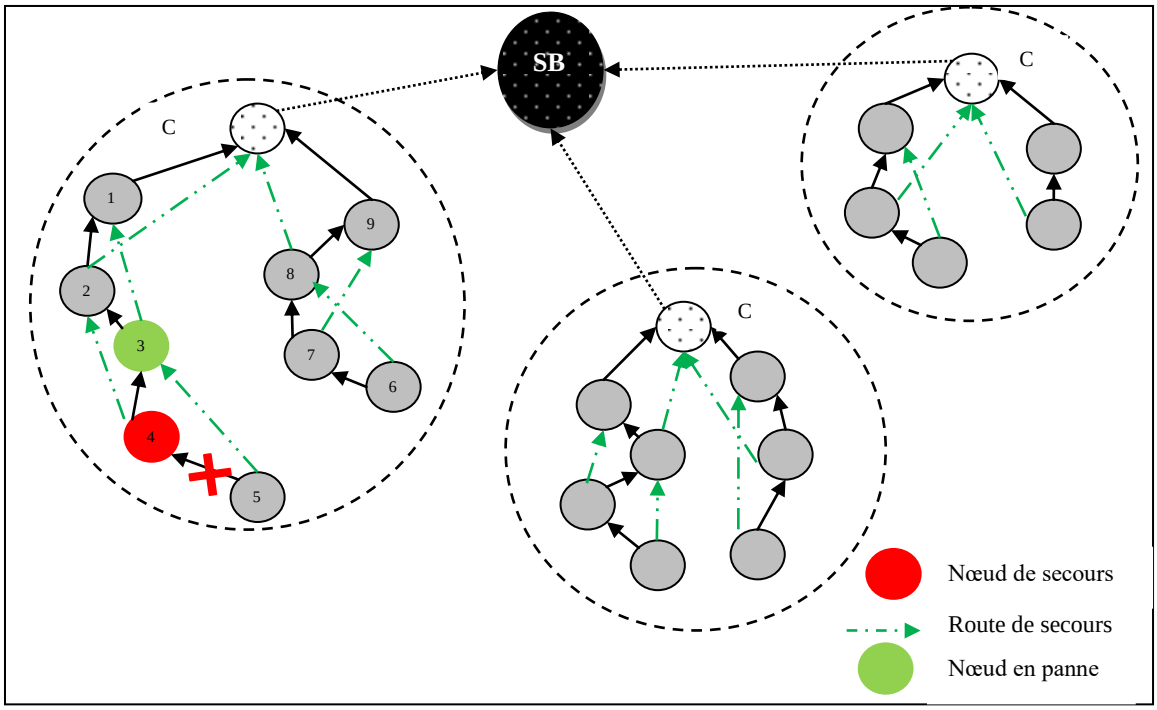



**Figure 5. 10.** Schema de recouvrement de la panne des noeuds

### *c. Recouvrement de la panne des CHs :*

Lorsque le CH tombe en panne, le nœud qui se trouve à un seul saut du CH qui a le plus grand degré d'énergie et un RSSI élevé se charge d'acheminer les données directement vers la SB comme le montre la figure 5.11. Ceci permet d'un côté d'éviter la perte des données du cluster et d'un autre côté d'éviter l'épuisement rapidement de l'énergie du nœud qui se charge d'acheminer les données vers la SB.

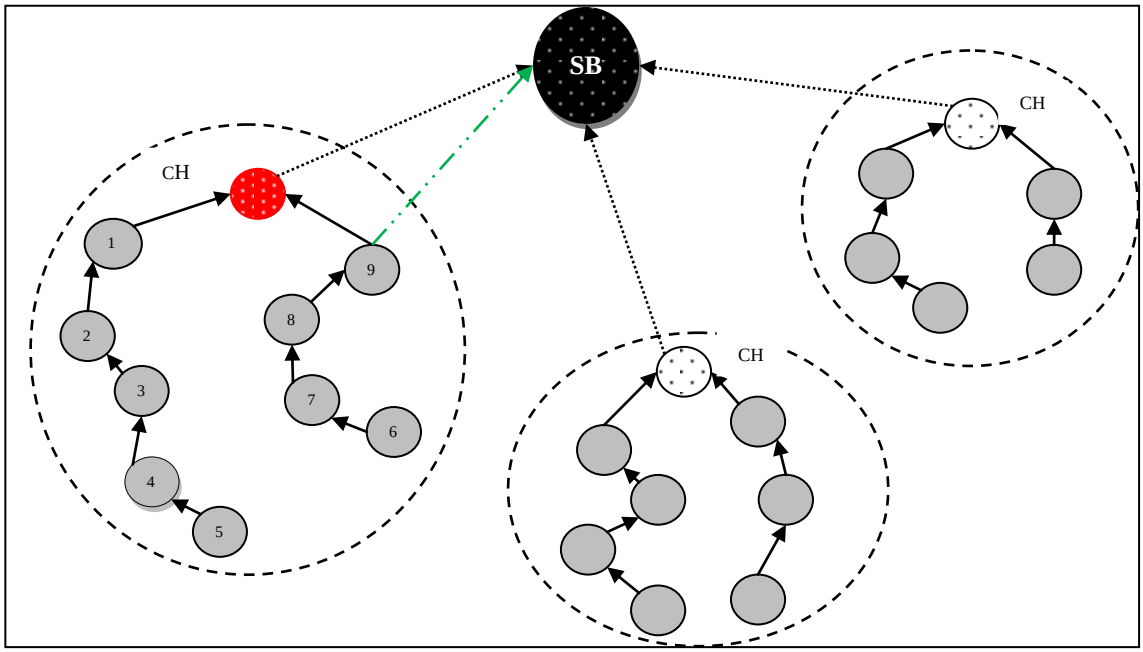



**Figure 5. 11.**Schema de recouvrement de la panne des CHs

Cette stratégie est plus économique en termes d'énergie puisque la consommation d'énergie est proportionnelle à la distance entre la Station de base et le nœud. Le nœud le plus éloigné de la Station de base consomme plus d'énergie.

## 5.6 Évaluation des performances du protocole FT-HEEP

Cette section est dédiée à la discussion des différents résultats obtenus après la simulation de notre nouveau protocole FT-HEEP. La simulation de notre nouveau protocole constitue la partie la plus importante de notre travail puisque on peut prouver les améliorations effectuées en termes tolérance aux pannes. L'analyse des performances de notre algorithme de routage est évaluée à l'aide du simulateur de réseau NS 2.34. Les résultats fournis par la simulation seront comparés aux autres protocoles à la fin de cette section.

### 5.6.1 Présentation du simulateur NS-2

***L***e simulateur ***NS-2*** [71,72] *(Network Simulator 2)* est un simulateur à événements discrets orienté objet qui permet d'exécuter tout type de scénarios sur des topologies définies par l'utilisateur. Il permet à l'utilisateur de définir un réseau et de simuler des communications entre les nœuds de ce réseau.

*NS2* utilise le langage OTCL *(Object Tools Command Language),* dérivé objet de TCL qui est un langage de programmation dont le but est de passer des commandes à des programmes interactifs tels que des éditeurs de texte, des débogueurs et des interpréteurs Shell. À travers ce langage, l'utilisateur décrit les conditions de la simulation : topologie du réseau, caractéristiques des liens physiques, protocoles utilisés, communications... La simulation doit d'abord être saisie sous forme de fichier texte que NS utilise pour produire un fichier trace contenant les résultats. Des outils périphériques permettent l'animation du réseau *(NAM : Network Animator)* ou la conversion vers d'autres outils *(comme par exemple xgraph pour dessiner des courbes).*

### 5.6.2 Avantages et limites de la simulation

L'utilisation de la simulation présente plusieurs avantages. D'une part, la simulation est peu couteuse, car les simulateurs existants sont majoritairement gratuits et Open Source. D'autre part, le temps de développent et de simulation est très inférieur comparé aux implémentations réelles. En outre, le simulateur permet d'accélérer le temps et de prédire le comportement du système étudié. À titre d'exemple, on peut simuler la durée de vie des capteurs en un temps très court, vu que le temps lui-même est virtuel. Aussi, le simulateur peut être un outil d'évaluation complémentaire aux expérimentations réelles, car dans certains cas de figure, il est difficile d'avoir un grand nombre de nœuds réels, d'où l'intérêt de la simulation qui permet de faire le passage à l'échelle par une simple modification du script de simulation.

On parle dernièrement d'un nouveau paradigme dans lequel le simulateur ne joue pas seulement le rôle d'un outil d'évaluation, mais aussi sera une composante qui participe de manière active à la réalisation de quelques tâches dans le système réel [77]. Par exemple, en se basant sur les résultats obtenus du réseau réel, le simulateur peut créer les tables de routage et distribuer les changements aux nœuds de ce réseau.

Malgré tous ces avantages, les outils de simulation présentent quelques limites. Le plus grand inconvénient étant lié aux modèles des protocoles et leurs degrés de correspondance à la réalité (p. ex. propagation des ondes radio). En effet, le modèle utilisé pour simuler un protocole est un modèle théorique qui a recours à plusieurs simplifications. Une étude a été menée dans [76] dans l'optique de voir la pertinence des résultats de simulation comparés aux implémentations réelles dans les réseaux sans fil. Cette étude montre que les résultats de simulation peuvent être proches des résultats réels dans certaines conditions, surtout si le choix du modèle de la couche physique et de ses paramètres (plus particulièrement, le modèle de propagation) est correct. Mais, le simulateur peut ne pas être adapté dans d'autres contextes. Ainsi, les résultats obtenus par simulation peuvent donner une estimation des performances, mais chaque protocole doit être validé sur des équipements réels afin d'obtenir des résultats plus précis et plus réalistes.

## 5.6.3. Environnement de simulation

L'analyse des performances de notre protocole est évaluée à l'aide du simulateur de réseau NS2.34 . Les résultats fournis par la simulation sont comparés au protocole HEEP et aux autres protocoles avec tolérance aux pannes. Dans cette simulation, notre modèle d'expérimentation est établi sur 100 nœuds répartis aléatoirement sur une surface carrée de 100x100 m² comme montré dans la figure 5.12. Nous assumons que tous les nœuds ont une position fixe durant toute la période de simulation.

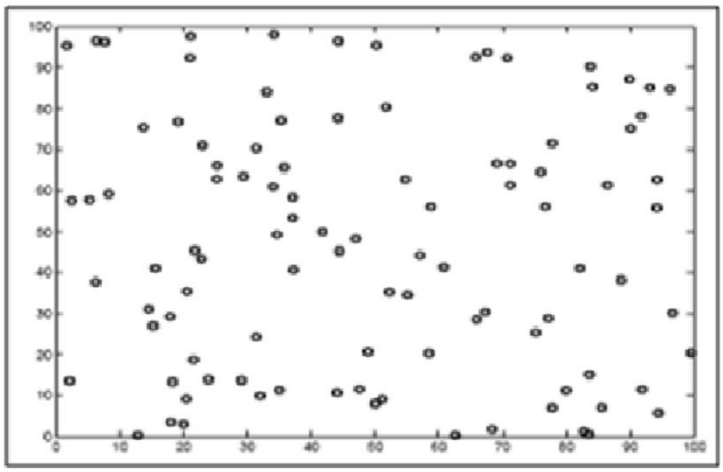



**Figure 5. 12.** Le modèle d'expérimentation.

Notre modèle de simulation utilise les paramètres résumés dans le tableau ci-dessous :

| Paramètre | Valeur |
|---|---|
| La surface du réseau | 100*100 m² |

| | |
|---|---|
| La localisation de la SB | (20,175) |
| Le nombre de noeuds | 100 |
| Nombre de clusters | 5 |
| L'énergie intiale des noeuds | 2 Joules |
| La taille du paquet de données | 512 bytes |
| Le modèle de trafic | CBR |
| Le modèle de propagation radio | Free Space |

**Table 5. 1.** *Les paramètres de simulation*

La station de base est positionnée à 75 mètres par rapport au nœud le plus proche (X=20, Y=175). La largeur de bande de transmission est initialisée à 1Mbps. Tous les nœuds du réseau commencent la simulation avec une énergie initiale de 2 J et une quantité de données illimitées à transmettre à la station de base. De plus, l'énergie de la station de base est considérée comme illimitée, la taille d'un paquet de données est égale à 512 Bytes, avec une entête de paquet mesurant 25 Bytes.

## 5.6.4 Métriques de performances

Afin d'analyser les performances de notre protocole, nous considérons les métriques suivantes :

- Le nombre de nœuds vivants
- Le PDR (Packet delivery ratio): Mesure le pourcentage de paquets de données générés par les noeuds sources et ceux reçus avec succès par la SB .
- Le débit : c'est le nombre de bits reçus par unité de temps
- Le taux de réception des données
- L'énergie consommée : La quantité d'énergie consommée par les nœuds capteurs du réseau.

### 5.6.4.1 Résultats de la simulation

#### *a. Packet delivery ratio (PDR)*

Le PDR mesure le pourcentage de paquets de données générés par les noeuds sources et ceux reçus avec succès par la SB. Il est calculé comme suit :

$$PDR = \left({}^{T_R}/_{T_S}\right) * 100\%$$

Ou: $T_S$ : est le nombre total de paquets de données envoyés par le nœud source.

$T_R$: est le nombre total de paquets de données reçus par la SB.

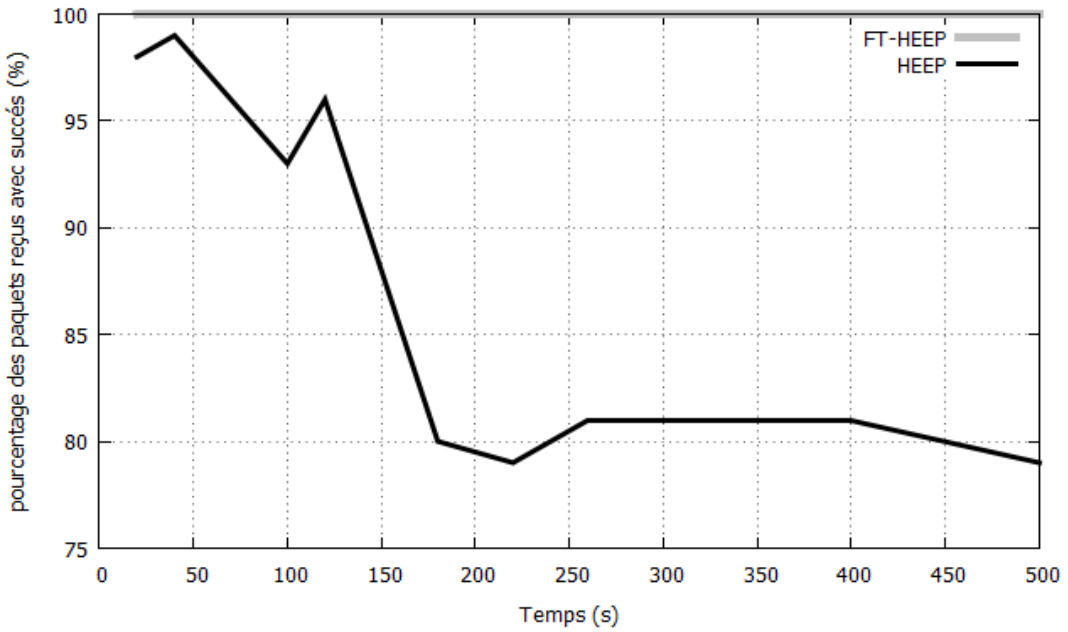



**Figure 5. 13.** Effet du temps de simulation sur le PDR

La figure 5.13 montre clairement que FT-HEEP a un meilleur taux de transmission réussie en présence de pannes des nœuds défaillants ou du CH. Ce qui prouve que contrairement à HEEP, FT-HEEP est tolérant aux pannes.

### *b. L'énergie consommée*

L'énergie du nœud est une ressource importante dans les réseaux de capteurs sans fil. Les noeuds fonctionnent à l'aide de piles qui ne sont généralement pas rechargeables. L'énergie est utilisée pour transmettre et recevoir des informations. Afin de démontrer l'efficacité énergétique de notre protocole, nous avons mesuré l'énergie consommée par les nœuds du réseau, ce qui nous a donné les résultats tel qu'illustré dans la figure 5.14. Le protocole FT-HEEP consomme un peu plus d'énergie que le protocole HEEP vu le nombre de messages échangées entre les nœuds pour assurer la tolérance aux pannes. Les résultats montrés dans la figure 5.15 confirment que notre protocole est plus efficace énergétiquement par rapport au protocole EF-LEACH ce qui prolonge la durée de vie du réseau.

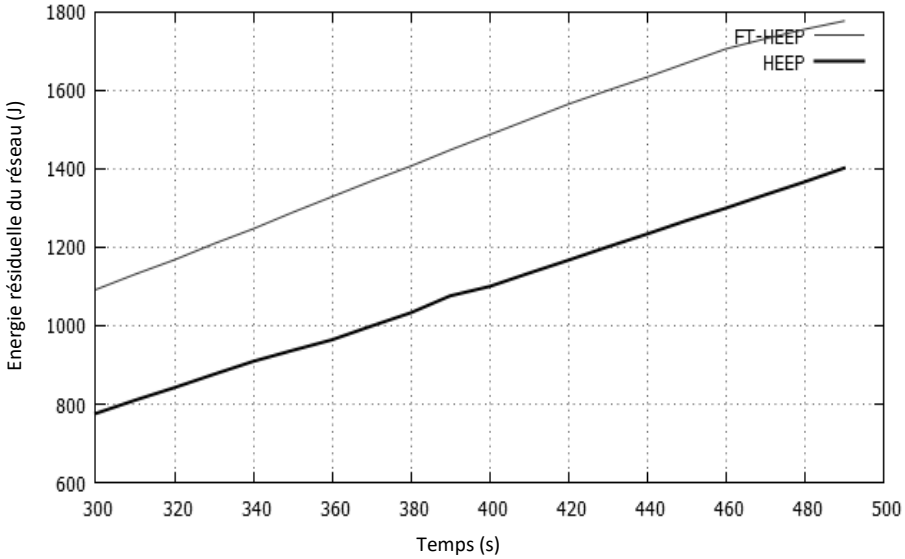



**Figure 5. 14.**Energie consommée par HEEP et FT-HEEP

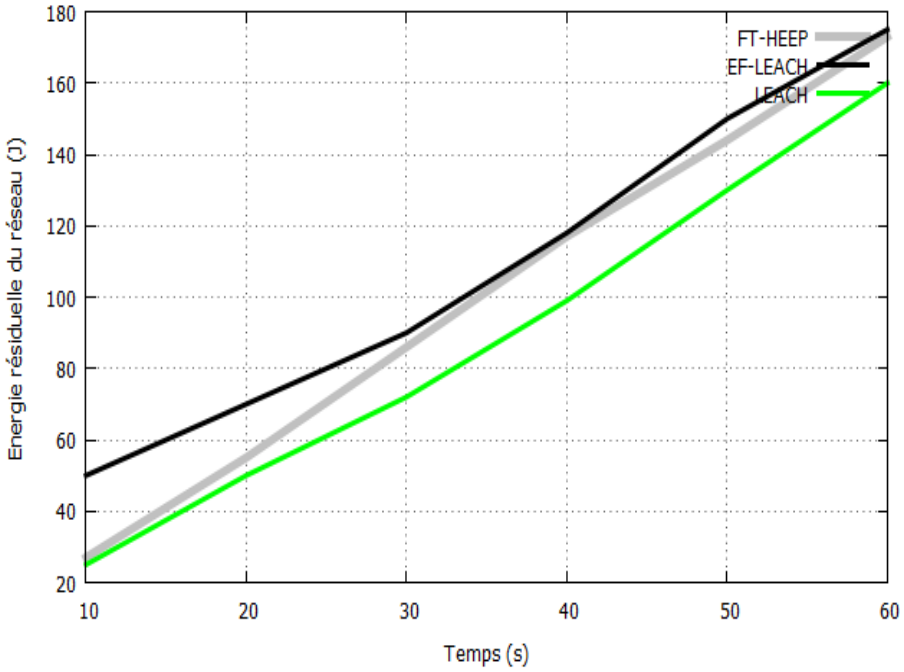



**Figure 5. 15.**Energie consommée par EF-LEACH, LEACH et FT-HEEP

### *c. Le débit*

Le débit mesure la quantité de données transmise par unité de temps. Il est calculé comme suit :

$$TP = \frac{le\ nombre\ de\ données\ reçues\ par\ la\ destination * 8}{T_G - T_F}$$

Ou : $TP$ : est le débit exprimé en bit/s, $T_G$ : est le temps de fin de la simulation et $T_F$ : est le temps de réception du premier paquet de données par la destination.

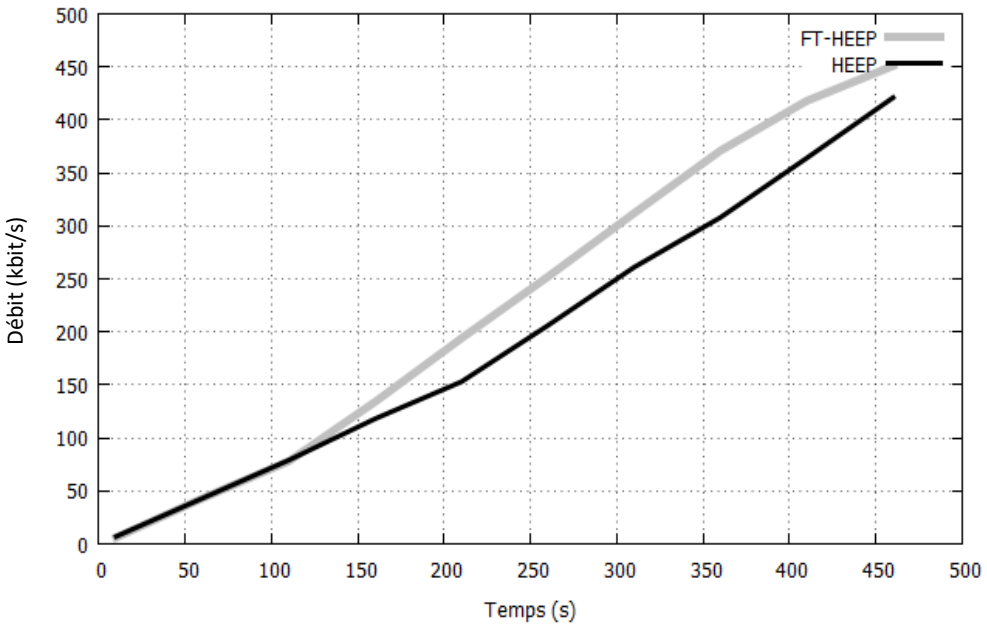



**Figure 5. 16.**Le débit mesuré en fonction du temps de la simulation

La figure 5.16 montre la variation du débit en fonction du temps de simulation pour les deux protocoles HEEP et FT-HEEP. Les résultats obtenus montrent que le débit de notre nouveau protocole FT-HEEP est plus élevé que celui du protocole HEEP, car FT-HEEP implémente un mécanisme de gestion des pannes pour éviter la perte de paquets.

### *d. Le taux de réception des données*

La figure 5.17 montre le taux de réception des données pour HEEP et FT-HEEP, en effet nous pouvons constater que les performances du protocole proposé sont meilleures que celle du protocole intiale ce qui prouve que notre nouveau protocole assure que les données arrivent à la station de base même en cas de panne de certains nœuds ou du CH.

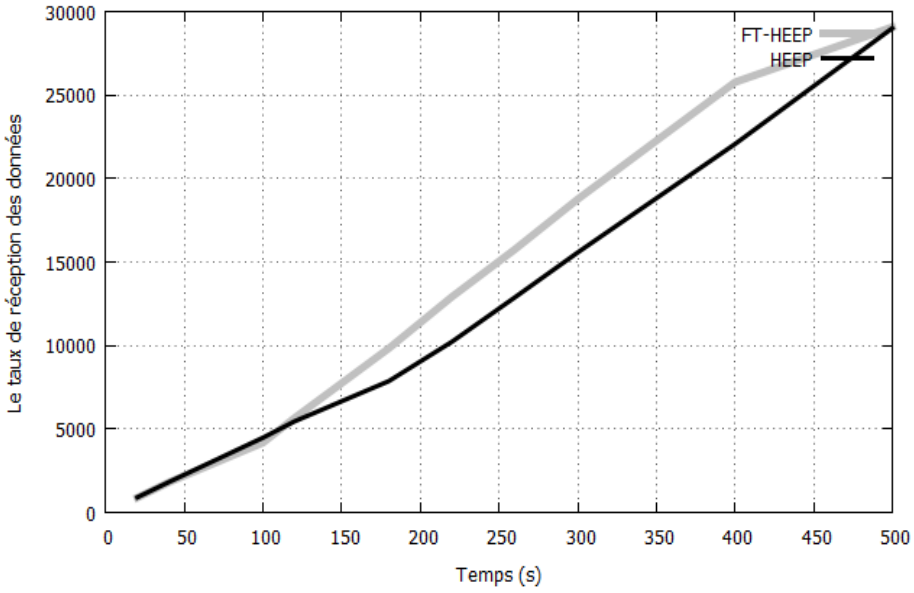



**Figure 5. 17.**Le taux de réception des données en fonction du temps de la simulation

### *e. Le nombre de nœuds vivants*

Comme illustré dans la figure 5.18, les resultats montrent que dans notre nouvelle solution les noeuds meurent plus rapidement comparé au protocole HEEP et ceci est principalement du à l'introduction des mécanismes pour la détection et le recouvrement des pannes.

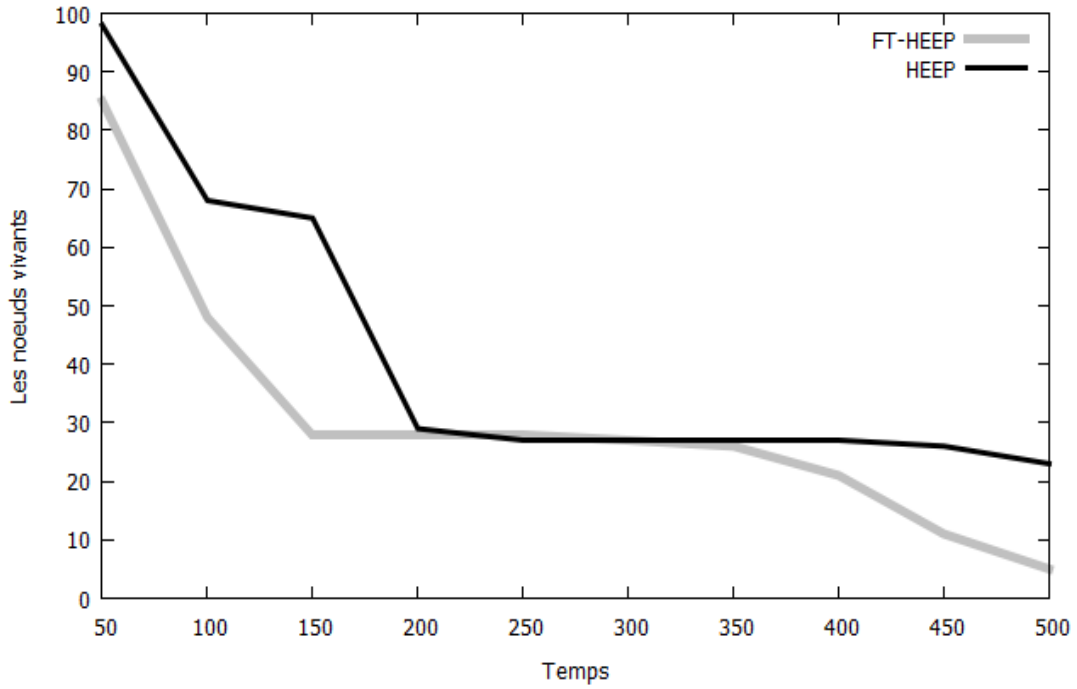



**Figure 5. 18.** Le nombre de nœuds vivants en fonction du temps de la simulation.

Afin de comparer les performances de notre protocole, nous avons mesuré le nombre de nœuds vivants pour les protocoles EF-LEACH, FTEAM, LEACH et HEEP ce qui nous a donné les résultats tel qu'illustré dans la figure 5.19. Le protocole FT-HEEP est meilleur que le protocole LEACH en terme de vivassité des nœuds, mais comparé à FTEAM, EF-LEACH et HEEP, les nœuds meurent plus rapidement vu que notre protocole est le seule qui introduit un mécanisme pour la détection et le recouvrement des pannes pour les nœuds et les CHs.

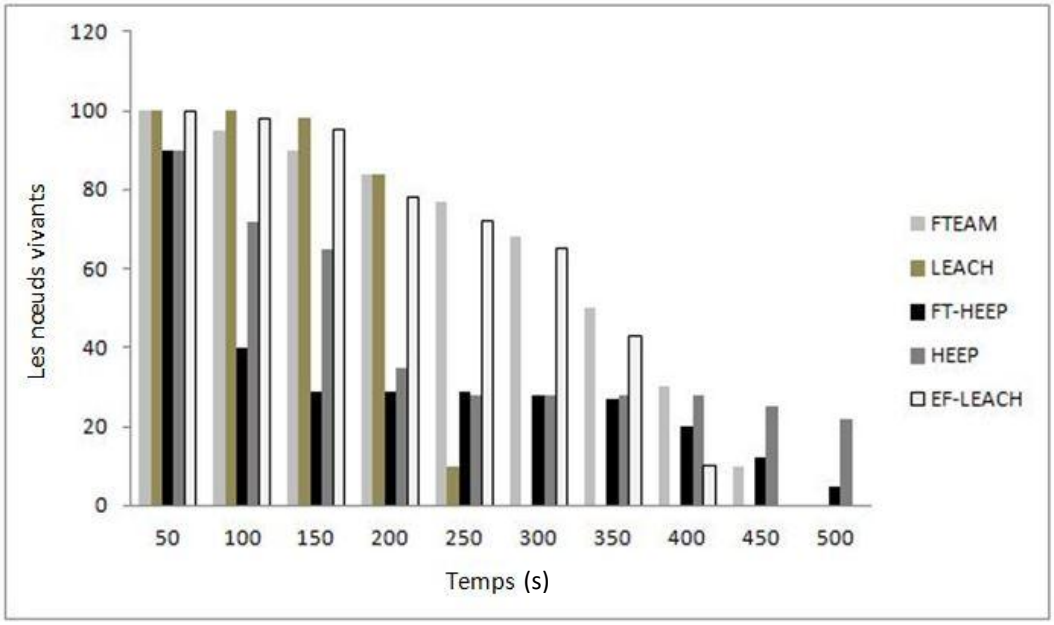



**Figure 5. 19.** Le nombre de nœuds vivants pour les protocoles FTEAM, LEACH, FT-HEEP, HEEP et EF-LEACH en fonction du temps de la simulation.

### f. Etude comparative

Pour conclure cette partie des résultats, nous les avons résumés dans le tableau ci-après :

| | **HEEP** | **FT-HEEP** | **FTEAM** | **LEACH** | **EF-LEACH** |
|---|---|---|---|---|---|
| **La tolérance aux pannex** | Non | Oui | Oui | Non | Oui |
| **La consommation d'énergie** | Très bonne | Proche de HEEP | Meilleure que LEACH | bonne | Meilleure que LEACH |
| **Détection des nœuds en panne.** | Non | Oui | Non | Non | Non |
| **Détection des CHs en panne** | Non | Oui | Oui | Non | Oui |
| **Les couches** | La couche réseau | Physique, MAC et la couche réseau | La couche réseau | La couche réseau | La couche réseau |
| **Cross Layers** | Non | Oui | Non | Non | Non |
| **Les nœuds vivants** | Meilleure que LEACH | inférieure à HEEP | Supérieure à LEACH | Faible | Meilleure que LEACH |

**Table 5. 2** Comparaison entre les différents protocoles.

## 5.7 Conclusion

Dans ce chapitre, nous avons décrit et discuté la solution cross layer que nous avons proposé pour assurer la tolétance aux pannes dans les réseaux de capteurs sans fil. Dans notre proposition le

problème de tolérance aux pannes est abordé avec une nouvelle stratégie dans laquelle l'approche cross-layer est pleinement exploitée. Notre approche consiste à produire un nouveau protocole cross-layer qui exploite l'interaction entre les trois premières couches du modèle OSI (couche physique, couche MAC, couche réseau) afin d'améliorer les performances du réseau et de garantir un certain niveau de tolérance aux pannes tout en choisissant des liens de bonne qualité et moins congestionnés.

Les résultats de l'évaluation montrent que notre protocole FT-HEEP présente une bonne solution pour assurer un bon fonctionnement du réseau même en cas de pannes par rapport à d'autres protocoles comme EF-LEACH, LEACH, FTEAM et HEEP.

# CONCLUSION GENERALE

*Durant ces dernières décennies, les réseaux de capteurs sans fil ont réussi à conquérir de nombreux secteurs d'activité et cela grâce aux avancées fulgurantes dans les domaines de la micro-électronique et de l'électromécanique. Cet important essor de la technologie des capteurs sans fil, fait que cette dernière attire aujourd'hui un très grand monde de chercheurs, aussi bien dans les universités que dans l'industrie. Cependant, ces réseaux, du fait de plusieurs paramètres intrinsèques notamment la tolérance aux pannes, font face à de nombreux défis.*

*L'assurance de la tolérance aux pannes dans le domaine des réseaux de capteurs sans fil représente un défi de recherche très important, plus particulièrement dans ce type de réseaux qui sont caractérisés par les contraintes des nœuds capteurs, et dans lesquels l'économie d'énergie est une question axiale.*

*Dans cette thèse, nous nous sommes intéressés à cette problèmatique, nous avons développé un protocole de routage tolérant aux pannes nommé FT-HEEP (Fault Tolerant Hybrid Energy Efficiency Protocol). Ce dernier exploite une architecture cross-layer basée sur une interaction entre les trois premières couches de la pile protocolaire à savoir la couche réseau, la sous couche MAC et la couche physique. Au niveau de la couche physique, l'évaluation de la qualité du lien est effectuée, cette information est utilisée par la couche réseau pour sélectionner le nœud ou le CH de secours qui sera utilisé pour acheminer les informations en cas de panne. Par ailleurs, ce protocole repose sur la coopération de la couche MAC afin d'éviter les zones congestionnées. Les résultats expérimentaux démontrent que notre solution offre un bon niveau de performances en termes de tolérance aux pannes*

*Bien que notre nouveau protocole réponde à la problématique de tolérance aux pannes, il n'est pas encore optimal. Il existe plusieurs points qui peuvent être améliorés en perspective. Dans un premier temps, nous comptons tester son passage à l'échelle étant donné que nous l'avons testé sur un réseau contenant uniquement cent nœuds capteurs. Il sera aussi interessant de le tester et l'adapter au cas où les nœuds capteurs sont hétérogènes ou encore mobiles étant donné que l'hétérogéneité et la mobilité sont deux paramètres très importants dans les RCSF. De plus, nous pensons qu'une optimisation basée sur des heuristiques peut améliorer sa consommation d'énergie afin d'éviter que les noeuds meurent rapidement.*

## A. Références Bibliographiques


[1] A.Bharathidasan, V.Anad Sau Ponduru,"Sensor networks : An overview", département d'informatique de Californie.

[2] Kacimi, R. (2009). *Techniques de conservation d'énergie pour les réseaux de capteurs sans fil* (Doctoral dissertation).

[3] L.Raileanu,F.Nastaran, " les réseaux de senseurs ", haute école d'ingénierie et de gestion du Couton de Vaud, 10/01/2006. »

[4] Lehsaini, M. (2009). *Diffusion et couverture basées sur le clustering dans les réseaux de capteurs: application à la domotique* (Doctoral dissertation, Université de Franche-comté. UFR des sciences et techniques).

[5] Heinzelman, W. R., Kulik, J., & Balakrishnan, H. (1999, August). Adaptive protocoles for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking* (pp. 174-185). ACM.

[6] Intanagonwiwat, C., Govindan, R., & Estrin, D. (2000, August). Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (pp. 56-67). ACM.

[7] Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000, January). Energy-efficient communication protocole for wireless microsensor networks. In *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on* (pp. 10-pp). IEEE.

[8] Mhatre, V., & Rosenberg, C. (2004). Design guidelines for wireless sensor networks: communication, clustering and aggregation. *Ad hoc networks*, *2*(1), 45-63.

[9] Ye, M., Li, C., Chen, G., & Wu, J. (2005, April). EECS: an energy efficient clustering scheme in wireless sensor networks. In *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*(pp. 535-540). IEEE.

[10] Lee, G., Kong, J., Lee, M., & Byeon, O. (2005). A cluster-based energy aware routing protocole for sensor networks. In *Parallel and Distributed Computing and Systems Proceedings of the 17 th IASTED International Conference*.

[11] Muruganathan, S. D., Ma, D. C., Bhasin, R. I., & Fapojuwo, A. O. (2005). A centralized energy-efficient routing protocole for wireless sensor networks. *IEEE Communications Magazine*, *43*(3), S8-13.

[12] Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocole architecture for wireless microsensor networks. *IEEE Transactions on wireless communications*, *1*(4), 660-670.

[13] Lindsey, S., & Raghavendra, C. S. (2002). PEGASIS: Power-efficient gathering in sensor information systems. In *Aerospace conference proceedings, 2002. IEEE* (Vol. 3, pp. 3-3). IEEE.

[14] He, T., Stankovic, J. A., Lu, C., & Abdelzaher, T. (2003, May). SPEED: A stateless protocole for real-time communication in sensor networks. In *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on* (pp. 46-55). IEEE.

[15] Braginsky, D., & Estrin, D. (2002, September). Rumor routing algorthim for sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* (pp. 22-31). ACM.

[16] Curiac, D. I., Volosencu, C., Pescaru, D., Jurca, L., & Doboli, A. (2009). Redundancy and its applications in wireless sensor networks: A survey. *WSEAS Transactions on Computers*, *8*(4), 705-714.

[17] Taleb, A. A., Pradhan, D. K., & Kocak, T. (2009, June). A technique to identify and substitute faulty nodes in wireless sensor networks. *Third International Conference on Sensor Technologies and Applications,SENSORCOMM'09.* (pp. 346-351). IEEE.

[18] Saleh, I., Agbaria, A., & Eltoweissy, M. (2006, September). In-network fault tolerance in networked sensor systems. In *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks* (pp. 47-54). ACM.

[19] Ramanathan, N., Chang, K., Kapur, R., Girod, L., Kohler, E., & Estrin, D. (2005, November). Sympathy for the sensor network debugger. In *Proceedings of the 3rd international conference on Embedded networked sensor systems* (pp. 255-267). ACM.

[20] Staddon, J., Balfanz, D., & Durfee, G. (2002, September). Efficient tracing of failed nodes in sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* (pp. 122-130). ACM.

[21] Tanachaiwiwat, S., Dave, P., Bhindwale, R., & Helmy, A. (2003, November). Poster abstract secure locations: routing on trust and isolating compromised sensors in location-aware sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems* (pp. 324-325). ACM.

[22] I-Iarte, S., Rahmanl, A., & Razeeb, K. M. (2005). Fault tolerance in sensor networks using self-diagnosing sensor nodes.

[23] Koushanfar, F., Potkonjak, M., & Sangiovanni-Vincentell, A. (2002). Fault tolerance techniques for wireless ad hoc sensor networks. In *Sensors, 2002. Proceedings of IEEE* (Vol. 2, pp. 1491-1496). IEEE.

[24] Chen, J., Kher, S., & Somani, A. (2006, September). Distributed fault detection of wireless sensor networks. In *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks* (pp. 65-72). ACM.

[25] Lee, M. H., & Choi, Y. H. (2007, October). Distributed diagnosis of wireless sensor networks. In *TENCON 2007-2007 IEEE Region 10 Conference* (pp. 1-4). IEEE.

[26] Xiang-hua, X., Biao, Z., & Jian, W. (2009, December). Tree topology based fault diagnosis in wireless sensor networks. In *Wireless Networks and Information Systems, 2009. WNIS'09. International Conference on* (pp. 65-69). IEEE.

[27] Ding, M., Chen, D., Xing, K., & Cheng, X. (2005, March). Localized fault-tolerant event boundary detection in sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE* (Vol. 2, pp. 902-913). IEEE.

[28] Estrin, D., Govindan, R., Heidemann, J., & Kumar, S. (1999, August). Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking* (pp. 263-270). ACM.

[29] Tai, A. T., Tso, K. S., & Sanders, W. H. (2004, June). Cluster-based failure detection service for large-scale ad hoc wireless network applications. In *Dependable Systems and Networks, 2004 International Conference on* (pp. 805-814). IEEE.

[30] Akbari, A., Dana, A., Khademzadeh, A., & Beikmahdavi, N. (2011). Fault detection and recovery in wireless sensor network using clustering. *International Journal of Wireless & Mobile Networks (IJWMN)*, *3*(1), 130-138.

[31] Gupta, G., & Younis, M. (2003, March). Fault-tolerant clustering of wireless sensor networks. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE* (Vol. 3, pp. 1579-1584). IEEE.

[32] Koushanfar, F., Potkonjak, M., & Sangiovanni-Vincentell, A. (2002). Fault tolerance techniques for wireless ad hoc sensor networks. In *Sensors, 2002. Proceedings of IEEE* (Vol. 2, pp. 1491-1496). IEEE.

[33] Clouqueur, T., Saluja, K. K., & Ramanathan, P. (2004). Fault tolerance in collaborative sensor networks for target detection. *IEEE transactions on computers*, *53*(3), 320-333.

[34] Marti, S., Giuli, T. J., Lai, K., & Baker, M. (2000, August). Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (pp. 255-265). ACM.

[35] Lee, W. L., Datta, A., & Cardell-Oliver, R. (2006). Winms: Wireless sensor network-management system, an adaptive policy-based management for wireless sensor networks.

[36] Kulothungan, K., Jothi, J. A. A., & Kannan, A. (2011). An adaptive fault tolerant routing protocole with error reporting scheme for wireless sensor networks. *European Journal of Scientific Research*, *16*(1), 19-32.

[37] Che-Aron, Z., Al-Khateeb, W., & Anwar, F. (2010). The enhanced fault-tolerance mechanism of aodv routing protocole for wireless sensor network. *International Journal of Computer Science and Network Security, IJCSNS*, *10*, 41-50.

[38] Qiu, M., Liu, J., Li, J., Fei, Z., Ming, Z., & Sha, E. H. (2011, August). A novel energy-aware fault tolerance mechanism for wireless sensor networks. In *Proceedings of the 2011 IEEE/ACM International Conference on Green Computing and Communications* (pp. 56-61). IEEE Computer Society.

[39] Jain, N., Vokkarane, V. M., & Wang, J. (2008, May). Performance analysis of dual-homed fault-tolerant routing in wireless sensor networks. In *Technologies for Homeland Security, 2008 IEEE Conference on* (pp. 474-479). IEEE.

[40] Lehsaini, M., & Hellel, C. T. (2012, December). A novel cluster-based fault-tolerant scheme for wireless sensor networks. In *Microelectronics (ICM), 2012 24th International Conference on* (pp. 1-4). IEEE.

[41] Karim, L., & Nasser, N. (2011, June). Energy efficient and fault tolerant routing protocole for mobile sensor network. In *Communications (ICC), 2011 IEEE International Conference on* (pp. 1-5). IEEE.

[42] Aliouat, Z., & Aliouat, M. (2013). Improving Wireless Sensor Networks Robustness through Multi-level Fault Tolerant Routing Protocole. In *Modeling Approaches and Algorithms for Advanced Computer Applications* (pp. 115-124). Springer International Publishing.

[43] Ajay, A., Tarasia, N., Dash, S., Ray, S., & Swain, A. R. (2011). A Dynamic Fault Tolerant Routing Protocole for Prolonging the Lifetime of Wireless Sensor Networks. *International Journal of Computer Science and Information Technologies*, *2*(2), 727-734.

[44] Ajay, N. T., Dash, S., & Ray, S. ARSwain,"Protocole de routage tolérant aux fautes multi-niveaux avec des horaires du sommeil (FMS) pour les réseaux de capteurs sans fil ". *European Journal of Scientific Research*, *55*(1)..

[45] Guowei Wu, ChiLin, Feng Xia, Lin Yao, il Zhang et Liu Bing «Saut dynamique en temps réel Fault-Tolerant protocole de routage pour les réseaux de capteurs sans fil» de la Fondation nationale des sciences naturelles de Chine par la concession numéro60703101 et n ° 60903153 (2010).

[46] Boukerche, A., Pazzi, R. W. N., & Araujo, R. B. (2004, October). A fast and reliable protocole for wireless sensor networks in critical conditions monitoring applications. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems* (pp. 157-164). ACM.

[47] Boukerche, A., Chatzigiannakis, I., & Nikoletseas, S. (2006). A new energy efficient and fault-tolerant protocole for data propagation in smart dust networks using varying transmission range. *Computer communications*, *29*(4), 477-489.

[48] Maryam Hezaveh, Zahra Shirmohammdi, Nezam Rohbani & Seyed Ghassem Miremadi. (2015). A Fault Tolerant and Energy Aware Mechanism for Cluster-based Routing Algorithm of WSNs. In the proceedings of IEEE International Symposium on Integrated Network Management (IM): (pp. 659-664).

[49] Mona M, Jamjoom.(2017). EEBFTC: Extended Energy Balanced with Fault Tolerance Capability Protocole for WSN. International Journal of Advanced Computer Science and Applications, 8(1), 253-258.

[50] Azharuddin, M., Kuila, P., Jana, P.K. (2015). Energy efficient fault tolerant clustering and routing algorithms for wireless sensor networks. Computers and Electrical Engineering, 41, 177- 190.

[51] Chakravarthi, R., Gomathy, C., Sebastian, S. K., Pushparaj, K., & Mon, V. B. (2010). A survey on congestion control in wireless sensor networks. *International Journal of Computer Science & Communication*, *1*(1), 161-164.

[52] Malar, R. T. (2010). Congestion control in wireless sensor networks based multi-path routing in priority rate adjustment technique. *proceedings of International Journal of Advanced Engineering & Applications*, 28-33.

[53] Paek, J., & Govindan, R. (2010). RCRT: Rate-controlled reliable transport protocole for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, *7*(3), 20.

[54] Zhou, Y., Lyu, M. R., Liu, J., & Wang, H. (2005, November). PORT: a price-oriented reliable transport protocole for wireless sensor networks. In *Software Reliability Engineering, 2005. ISSRE 2005. 16th IEEE International Symposium on* (pp. 10-pp). IEEE.

[55] Bian, F., Rangwala, S., & Govindan, R. (2007, June). Quasi-static centralized rate allocation for sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on* (pp. 361-370). IEEE.

[56] Wan, C. Y., Eisenman, S. B., & Campbell, A. T. (2011). Energy-efficient congestion detection and avoidance in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, *7*(4), 32.

[57] Wan, C. Y., Eisenman, S. B., & Campbell, A. T. (2003, November). CODA: Congestion detection and avoidance in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems* (pp. 266-279). ACM.

[58] Sankarasubramaniam, Y., Akan, Ö. B., & Akyildiz, I. F. (2003, June). ESRT: event-to-sink reliable transport in wireless sensor networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing* (pp. 177-188). ACM.

[59] Ee, C. T., & Bajcsy, R. (2004, November). Congestion control and fairness for many-to-one routing in sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (pp. 148-161). ACM.

[60] Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A., & Estrin, D. (2004). Habitat monitoring with sensor networks. *Communications of the ACM*, *47*(6), 34-40.

[61] Zhao, F., Shin, J., & Reich, J. (2002). Information-driven dynamic sensor collaboration. *IEEE Signal processing magazine*, *19*(2), 61-72.

[62] Joseph, A. D. (2005). Energy Harvesting Projects, Published by the IEEE CS and IEEE ComSoc.

[63] Nivor, F. (2009). *Architecture de communication pour les applications multimédia interactives dans les réseaux sans fil* (Doctoral dissertation, Université Paul Sabatier-Toulouse III).

[64] Bae, S. Y., Lee, S. K., & Park, K. W. (2013). Cross-layer QoS architecture with multipath routing in wireless multimedia sensor networks. *International Journal of Smart Home*, *7*(3), 219-226.

[65] Boubiche, D.E., & Bilami, A. (2011). HEEP (Hybrid Energy Efficiency Protocol) Based on Chain Clustering. International Journal of Sensor Networks, 10 (1/2), 25-35.

[66] Djebaili, Y., & Bilami, A. A Cross-Layer Fault Tolerant Protocol with Recovery Mechanism for Clustered Sensor Networks, International Journal of Distributed System and Technologies (IJDST), 9(1), (2018), DOI: 10.4018/IJDST.2018010104

[67] Boubiche, D. E. (2013) 'Une approche Inter-Couches (cross-layer) pour la Sécurité dans les RCSF', Université de batna, thèse de Doctorat en Sciences en Informatique

[68] Baccour, N., Koubaa, A., Mottola, L. et al. (2012) 'Radio link quality estimation in Wireless sensor networks: a survey', ACM Transactions on Sensor Networks, 8(4), article 34

[69] Tariq, S. (2005). 'MAC Algorithms in Wireless Networks', Master's thesis, Umea University", Sweden, www.cs.umu.se/education/examina/Rapporter/ShoaibTariq.pdf.

[70] Bildea Ana. (2013) 'Link Quality in Wireless Sensor Networks'. Other [cs.OH]. University of Grenoble, English. <NNT: 2013GRENM054>. <tel-01167272>.

[71] Fall, K. and Varadhan, K. (2002) ' NS Notes and Documentation'. The VINT Project.

[72] Altman, E. and Jimenez, T. (2003) 'NS Simulator for beginners'. Technical Report, Univ. Los Andes, Merida, Venezuela & Sophia-Antipolis, France.

[73] S Sahraoui, A Bilami, Compressed and distributed host identity protocol for end-to-end security in the IoT, Fifth International Conference on Next Generation Networks and Services (NGNS 2014), 28-30 May 2014, Casablanca, Morocco

[74] Berthomieu, B., &Menasche, M. (1983). An Enumerative Approach for Analyzing Time Petri Nets. IFIP Congress Series, Vol. 9, (pp. 41-46).

[75] IEEE Std. 802.11 (1999), Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ISO/IEC 8802-11:1999(E), IEEE Std. 802.11.

[76] Rachedi, A., Lohier, S., Cherrier, S. and Salhi, I. (2010) ' Wireless Network Simulators Relevance Compared to a Real Testbed in Outdoor and Indoor Environments ', in Proceedings of the 6th International Wireless Communications and Mobile Computing

Conference, New York, NY, USA, 2010, pp. 346–350.

[77] El Gholami, K. (2015) 'La gestion de la qualité de service temps-réel dans les réseaux de capteurs sans fil', université blaise pascal - Clermont II, thèse en cotutelle avec l'université chouaib doukkali - El Jadida, 29 May 2015.

[78] S Maamar, A Lamia, G Leila, B Azeddine, Etude des performances des protocoles de routage dans les réseaux mobiles ad-hoc, 4th International Conférence on Computer Integrated Manufacturing CIP'2007

## B. Références Web

[W1] http://en.wikipedia.org/wiki/Networktopology.

ANNEXE A

# PRESENTATION DE NS-2

NS-2 comme présenté dans cette thèse est un outil logiciel de simulation de réseaux informatiques à événements discrets. Il est principalement bâti avec les idées de la conception par objets, de ré-utilisabilité du code et de modularité. C'est un logiciel libre qu'on retrouve facilement sur Internet. Le logiciel est exécutable sous Linux, MAC et Windows (en utilisant CygWin). NS-2 au départ a été conçu pour faciliter l'étude de l'interaction entre les protocoles et le comportement d'un réseau à différentes échelles. Pour cela, il contient des bibliothèques pour la génération de topologies réseaux, des trafics, ainsi qu'un outil de visualisation tel que l'animateur réseau NAM (network animator). Il est maintenant un outil bien adapté aux réseaux à communications de paquets et à la réalisation de simulations de petite taille. Il contient les fonctionnalités nécessaires à l'étude des algorithmes de routage unipoint ou multipoint, des protocoles de transport, de session, de réservation, des services intégrés, des protocoles d'application comme HTTP. De plus le simulateur possède déjà une palette de systèmes de transmission, d'ordonnanceurs et de politiques de gestion de files d'attente pour effectuer des études de contrôle de congestion. On retrouve dans le tableau suivant les principaux composants disponibles dans NS par catégorie :

| | |
|---|---|
| **Application** | Web, FTP, Telnet, générateur de trafic (CBR,…) |
| **Transport** | TCP, UDP, RTP,… |
| **Routage** | DSDV, DSR, AODV, PUMA, TORA,Flooding, … |
| **Gestion de la file d'attente** | RED, DropTail, Token bucket. |
| **Discipline de service** | CBQ, SFQ, DRR, Fair queueing |
| **Système de transmission** | CSMA/CD, CSMA/CA, lien point à point. |

En combinant tous les composants, ces capacités ouvrent le champ à l'étude de nouveaux mécanismes au niveau des différentes couches de l'architecture réseau. NS est devenu l'outil de référence pour les chercheurs du domaine.

## Notions pour l'interpréteur

NS-2 est un langage écrit en C++, avec un interpréteur OTcl. Dans ce paragraphe nous allons donner les bases du langage Tcl, les principes de l'OTcl et les explications sur le mécanisme qui lie le C++ avec l'interpréteur Tcl.

# 1. Tcl (Tool Command Language)

Est un langage de commande comme le shell UNIX mais qui sert à contrôler les applications. Il offre des structures de programmation telles que les boucles, les procédures ou les notions de variables. Il y a deux principales façons de se servir de Tcl : soit comme un langage autonome interprété ou comme une interface applicative d'un programme classique écrit en C ou C++. Toutes les applications qui utilisent Tcl créent et utilisent un interpréteur Tcl. Cet interpréteur est le point d'entrée standard de la bibliothèque. L'application tclsh constitue une application minimale ayant pour but de familiariser un utilisateur au langage Tcl et ne comporte que l'interpréteur Tcl.

On retrouve cet interpréteur dans l'application NS. Une fois la commande "ns" tapée, l'application effectue l'initialisation des objets puis passe en mode interactif ou on peut alors commencer à entrer les commandes Tcl.

## 1.1. Concepts

Tcl est un langage non typé ou chaque commande consiste en un ou plusieurs mots séparés par des espaces ou des tabulations. Tous les mots sont des chaînes de caractères. Le premier mot de la commande est le nom de la commande, les autres mots sont les arguments passés à la commande. Chaque commande Tcl retourne le résultat sous forme d'une chaîne de caractères. Le caractère de "retour à la ligne" termine une commande et lance son interprétation. Le caractère de séparation de plusieurs commandes sur une même ligne est " ;".

A l'inverse du C ou C++, Tcl n'est pas un langage compilé, mais un langage interprété. Tcl évalue une commande en effectuant une analyse syntaxique et son exécution. L'analyse syntaxique consiste à identifier les mots et effectuer les substitutions. Durant cette étape, l'interpréteur ne fait que des manipulations de chaînes. Il ne traite pas la signification des mots. Pendant la phase d'exécution, l'aspect sémantique des mots est traité comme par exemple déduire du premier mot le nom de la commande, vérifier si la commande existe et appeler la procédure de cette commande avec les arguments. Le backslash (\) et les groupages permettent d'insérer des caractères spéciaux dans les mots et d'écrire des commandes sur plusieurs lignes.

**Exemple**

| | |
|---|---|
| set a 12 | affecte la valeur 12 à la variable a |
| expr 2 + 3 | calcule la valeur de l'expression 2 + 3 |
| puts Coucou | affiche Coucou sur la sortie standard |

# 2. OTcl

OTcl est une extension orientée objet de Tcl. Les commandes Tcl sont appelées pour un objet. En OTcl, les classes sont également des objets avec des possibilités d'héritage. Les correspondances avec le C++ sont :

- C++ a une unique déclaration de classe. En OTcl, les méthodes sont attachées à un objet ou à une classe.
- Les méthodes OTcl sont toujours appelées avec l'objet en préfixe.
- L'équivalent du constructeur et destructeur C++ en OTcl sont les méthodes init {} destroy{}
- L'identification de l'objet lui-même : this(C++), $self (OTcl). $self s'utilise à l'intérieur d'une méthode pour référencer l'objet lui-même. A la différence de C++, il faut toujours utiliser $self pour appeler une autre méthode sur le même objet. C'est à dire "$self xyz 5 " serait "this− > xyz(5)" ou juste "xyz(5)" en C++.
- L'héritage multiple est possible dans les deux langages.

## 3. Lien C++ et Tcl

Construire une application avec un interpréteur Tcl revient à inclure une bibliothèque Tcl qui définit les commandes de bases de Tcl dans l'application. Comme nous l'avons dit, l'interpréteur effectue l'analyse syntaxique et appelle la fonction C correspondant à la commande Tcl. Ajouter une commande Tcl consiste à établir un lien entre un mot et une fonction C. Le mot sera le nom de la commande Tcl. La fonction C est définie dans le code source de l'application. Au démarrage, l'application procède dans son main() aux initialisations nécessaires et passe la main à l'interpréteur. L'application passe en mode interactif : à chaque commande tapée par l'utilisateur, la fonction C correspondante est appelée afin de réaliser la commande demandée.