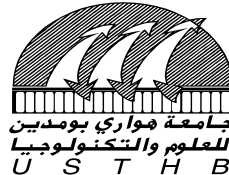


République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la technologie Houari Boumediene
(U.S.T.H.B)
BAB EZZOUAR, Alger



Faculté d'Electronique et de l'Informatique

Mémoire de Magister
En informatique
Option : informatique mobile (IM)

Thème

**Utilisation des cartes SOM pour le clustering
dans les réseaux de capteurs sans fil**

Réalisé par :
Melle. Fouzia Hamadi

Soutenue publiquement le 13/05/2010 devant le jury :

Mr . Abdelkader Belkhir, Professeur, USTHB.
Mr Nadjib Badache, Professeur, CERIST.
Mr Abdelkrim Abdelli, Docteur, USTHB.
Mr Mahfoud Benchaiba, Docteur, USTHB.

Président
Directeur de thèse
Examineur
Examineur

Promotion: 2005/2006

Remerciements

*Un grand merci et une grande reconnaissance à mon enseignant et promoteur Mr. **Badache** pour son énorme patience, sa disponibilité, son aide appréciable et ses conseils lumineux qu'il a toujours mis à mon égard afin de mener à bien ce modeste travail.*

*Je remercie Mr. **Belkhir** pour l'honneur qu'il m'accorde en présidant le jury de ce mémoire.*

*Merci à Mr. **Abdelli** et à Mr. **Benchai** pour avoir accepté de faire part du jury.*

Merci à tous.

Dédicaces

A la personne qui m'est la plus chère au monde « Maman » et à la mémoire du défunt « Papa » qui ont beau sacrifié et qui ont toujours donné de leur mieux pour m'éduquer et m'instruire pour enfin voir ce jour arriver.

*A mon cher mari **Rezki** qui, sans cesse, m'a donné la force et m'a encouragé afin d'aboutir mon projet et d'avancer dans ma vie.*

A mes sœurs et frères qui ont toujours été présents pour me soutenir et encourager.

A mes belles sœurs et à mes adorables petits neveux et nièces.

A Meriem, Selma, Lamis et toutes mes amies ainsi qu'à mes collègues dont la compagnie m'était un véritable plaisir.

Je dédie ce modeste travail.

Sommaire

Introduction générale.....	8
<i>Premier chapitre : Le clustering des réseaux de capteurs et le problème d'énergie.....</i>	<i>12</i>
I Architecture et terminologie.....	13
II Pile protocolaire	17
II.I Le niveau gestion d'énergie	17
II.II Le niveau gestion de mobilité	18
II.III Le niveau gestion des tâches	18
II.IV La couche application.....	18
II.V La couche transport	19
II.VI La couche réseau	19
II.VII La couche liaison de données.....	20
II.VIII la couche physique	20
III Critères de conception d'un protocole de routage.....	20
IV Modes de dissémination	22
IV.I Routage à base d'un saut :.....	22
IV.II Routage à base de plusieurs sauts :	22
IV.III Routage hiérarchique :	23
V Classes de protocoles de routage.....	24
V.I Les protocoles centrés données	24
V.II Les protocoles hiérarchiques	25
V.III Les protocoles géographiques	25
V.IV Les protocoles orientés qualité de service.....	25
VI Quelques protocoles de routage hiérarchique	26
VI.I Le protocole LEACH (Low Energy Adaptive Clustering Hierarchy)	26
VI.II PEGASIS (Power-Efficient Gathering in Sensor Information Systems).....	28
VI.III TEEN (Threshold sensitive Energy Efficient sensor Network protocol).....	30
VI.IV Le protocole GAF (Geographic Adaptive Fidelity)	31
VII Conclusion.....	32
<i>Deuxième chapitre : Les cartes auto-organisatrices de Kohonen (SOM).....</i>	<i>33</i>
I Introduction	34
II Domaines d'application.....	35
III Définitions	36
IV Cartes auto-organisatrices SOM (Self Organizing Map)	38
IV.I Modélisation et terminologie	38
IV.II Classes d'algorithmes d'optimisation de la fonction coût :	43
IV.II.1 Algorithmes d'optimisation non adaptative des cartes topologiques :.....	43
IV.II.2 Algorithmes d'optimisation adaptative des cartes topologiques et cartes topologiques évolutives :.....	50
IV.II.3 Algorithmes d'optimisation probabiliste :.....	53
V Conclusion.....	57
VI Quelques protocoles de routage à base de cartes SOM.....	59
VI.I Le protocole LEA2C (Low Energy Adaptive Connectionist Clustering).....	59
VI.II Le protocole CODA (Cluster-based self-Organizing Data Aggregation) [SangHak et al.05]	62
VII Synthèse	64
VIII Conclusion.....	65
<i>Troisième chapitre : SOBC (Self Organizing Based Clustering)</i>	<i>66</i>
I Introduction	67
II Problématique.....	68
III SOBC (Self Organizing Based Clustering).....	72

III.I	Personnalisation et paramétrage des cartes SOM.....	73
III.I.1	Liens entre les couches d'observations et de compétition	73
III.I.2	Dimension de la couche de compétition	74
III.I.3	Paramètre à optimiser.....	74
III.I.4	Initialisation des référents et du nombre d'itérations de l'algorithme d'auto-organisation	75
III.I.5	Définition du paramètre T et de sa loi de décroissance.....	77
III.I.6	Répartition du traitement de clustering	77
III.II	Principe de fonctionnement et fréquence d'exécution du clustering	79
III.II.1	Qu'est ce qui peut déclencher la restructuration des liens de clustering ?.....	79
III.II.2	Quelle est la portée du calcul du clustering ?.....	80
III.II.3	Quelle fonction d'affectation χ utiliser ?.....	80
III.II.4	Mise à jour des référents	80
III.II.5	Peut-on avoir une boucle en exécutant l'algorithme d'auto-organisation ?.....	81
III.II.6	Construction de la liste de voisinage ?.....	82
III.II.7	Qui exécute le calcul du clustering ?.....	82
III.III	L'algorithme d'auto-organisation SOBC	85
IV	Conclusion.....	86
	<i>Quatrième chapitre : Simulations et évaluation de performances</i>	87
I	Les métriques	88
I.I	Fréquence d'exécution du clustering.....	89
I.II	Le nombre de messages transmis pour le clustering	89
I.III	Concordance du clustering résultant avec les liens physiques réels	91
I.IV	Energie consommée et durée de vie des nœuds par rapport à la durée de vie globale du réseau.....	92
II	Conclusion.....	93
	Conclusion générale	94
	Bibliographie.....	98

Liste des Figures

Figure 1.	Architecture des réseaux capteurs.....	13
Figure 2.	Exemples de champs de captage.....	14
Figure 3.	Exemples de nœuds capteurs.....	14
Figure 4.	Les composants d'un nœud capteur.....	15
Figure 5.	Pile protocolaire dans les réseaux de capteurs.....	17
Figure 6.	Le modèle à un saut.....	22
Figure 7.	Le modèle multi-sauts.....	22
Figure 8.	Le modèle hiérarchique.....	23
Figure 9.	Le chaînage dans PEGASIS.....	28
Figure 10.	La collecte de données dans le protocole PEGASIS hiérarchique.....	29
Figure 11.	Exemple de grille virtuelle dans GAF.....	32
Figure 12.	Principe général de la modélisation.....	38
Figure 13.	Notion de distance dans une carte auto-organisatrice.....	40
Figure 14.	Fonction de voisinage à seuil.....	41
Figure 15.	Fonction de voisinage de type gaussien.....	41
Figure 16.	Principe général de la modélisation par carte auto-organisatrice.....	42
Figure 17.	Déroulement de l'algorithme des nuées dynamiques à T fixé.....	45
Figure 18.	Représentation de l'ordre induit par la carte topologique pour deux décroissances différentes..	46
Figure 19.	Carte topologique en deux dimensions.....	50
Figure 20.	Architecture à trois couches.....	54
Figure 21.	Deux clusterings successifs :SOM suivi de K-means.....	60
Figure 22.	Clustering géographique.....	62
Figure 23.	Re-clustering centré données.....	63
Figure 24.	Choix basé sur le maximum d'énergie.....	69
Figure 25.	Clustering résultant d'un algorithme de construction d'arbre.....	70
Figure 26.	Répartition des nœuds et liens de voisinage.....	77
Figure 27.	Suppression des liens correspondants aux nœuds non voisins.....	78
Figure 28.	Partitionnement de la carte.....	78
Figure 29.	Reconstruction du super cluster.....	84
Figure 30.	Nombre de nœuds vivants en fonction du temps pour LEA2C.....	92
Figure 31.	Nombre de nœuds vivants en fonction du temps pour SOBC.....	92

Liste des Tableaux

Tableau 1.	Différences entre les réseaux de capteurs et les réseaux Ad hoc	16
Tableau 2.	Synthèse des protocoles de routage	65

Introduction générale

Introduction générale

Les avancées récentes qu'a connu le domaine de communication sans fil et les technologies MEMS (Micro-Electro-Mecanical Systems) ont permis le développement des micro-composants qui intègrent des dispositifs de captage de données, de leur traitement et de communication sans fil dans un seul circuit à dimension réduite et avec un coût raisonnable [Akyildiz et al. 02].

Ces micro-composants sont appelés micro-capteurs, et leur évolution a engendré l'apparition des réseaux de capteurs (Sensor Networks).

La technologie des réseaux de capteurs est très prometteuse, la revue Business Week a souligné les opportunités sans précédent et les effets pratiques et pertinents que peuvent apporter ces réseaux dans beaucoup de domaines d'application à savoir le domaine militaire, sanitaire, civil et environnemental [Bus 99].

Les réseaux sensor sont basés sur le regroupement d'un grand nombre de micro-capteurs déployés de façon dense et qui opèrent de manière autonome, ils communiquent entre eux via des transmissions sans fil à courtes portées et ils collaborent pour l'accomplissement de l'analyse du domaine étudié.

Les nœuds captent, traitent et envoient les données aux nœuds collecteurs qui se chargent de l'agrégation des données et de leur fusion ainsi que la transmission des résultats aux nœuds qui ont interrogé le réseau de capteurs.

Les réseaux sensor constituent une sous classe des réseaux Ad hoc, ils sont caractérisés par plusieurs critères faisant leur particularité par rapport aux réseaux Ad hoc conventionnels à savoir [Khelladi, Badache 04] :

- Le nombre de nœuds est nettement plus grand dans les réseaux de capteurs que dans les réseaux Ad hoc.
- Les nœuds capteurs sont déployés de manière dense.
- Les nœuds capteurs sont plus exposés aux pannes.

- Les réseaux sensor utilisent les communications broadcast alors que la plupart des réseaux Ad hoc sont basés sur des communications point à point.
- Les nœuds capteurs ont des ressources plus limitées (énergie, puissance de calcul et mémoire).
- Les nœuds capteurs ne possèdent aucune identification (ID) globale telle que les adresses IP dans les réseaux Ad hoc.

Ces spécificités et contraintes des réseaux sensor font que les protocoles utilisés dans les réseaux Ad hoc ne peuvent s'appliquer de manière fiable et efficace. Par conséquent, des recherches intensives sont en cours d'élaboration, elles focalisent principalement sur la collaboration et la coordination entre les nœuds, l'agrégation et le traitement des données captées, tout en tenant compte des limitations en ressources de calcul et de communication et surtout en énergie qui présente la contrainte majeure dans les réseaux de capteurs.

Cependant, lors de la conception des différents protocoles opérationnels sur les réseaux de capteurs, il faut impérativement viser le but de la conservation d'énergie et de la préservation maximale de la durée de vie du réseau en entier. Pour ce faire, ces protocoles doivent toucher aux différentes couches de la carte réseaux, entre autres la couche réseaux qui doit permettre un routage optimal des données et garantir une connexion continue entre les nœuds et une durée de vie maximale du réseau.

Le clustering s'avère une technique de routage efficace pour l'optimisation de la consommation d'énergie. Les cartes auto-organisatrices constituent une méthode de clustering optimale basée sur plusieurs algorithmes, à savoir l'algorithme de Kohonen tel que leur exploitation dans le clustering des réseaux de capteurs est une approche prometteuse et elle réalise une amélioration considérable de la survivabilité du réseau en entier.

Dans notre travail, nous avons exploité les cartes topologiques de Kohonen, de manière complètement distribuée, afin de structurer notre réseau en clusters. Nous avons ainsi construis un protocole de routage à plusieurs niveaux d'hierarchisation. Les simulations ont montré une réduction importante de la consommation d'énergie. D'une part, ceci est réalisé grâce à l'utilisation des cartes de kohonen. D'une autre part, la définition de la nouvelle métrique qui représente « la fréquence d'exécution du processus de clustering » avait un apport considérable dans l'optimisation de la contribution des nœuds dans la tâche du calcul du clustering.

Le reste de ce document est organisé comme suit :

Le premier chapitre est une brève description des réseaux de capteurs (architecture, caractéristiques et spécifications) et des différents modes de routage et plus particulièrement, les protocoles de routage hiérarchiques.

Dans le deuxième chapitre, nous aborderons les cartes SOM et nous montrerons la possibilité de leur application au clustering des réseaux de capteurs. Nous expliquerons aussi quelques protocoles de routage à base de cartes auto-organisatrices et nous donnerons une synthèse de l'ensemble des protocoles étudiés.

Quant au troisième chapitre, il sera dédié à notre protocole de routage hiérarchique SOBC (Self Organizing Based Clustering).

Nous exposerons dans le quatrième chapitre les différentes métriques mesurées pour évaluer notre solution ainsi que les résultats de nos simulations.

Enfin, nous clôturons ce document par une conclusion générale et une liste bibliographique.

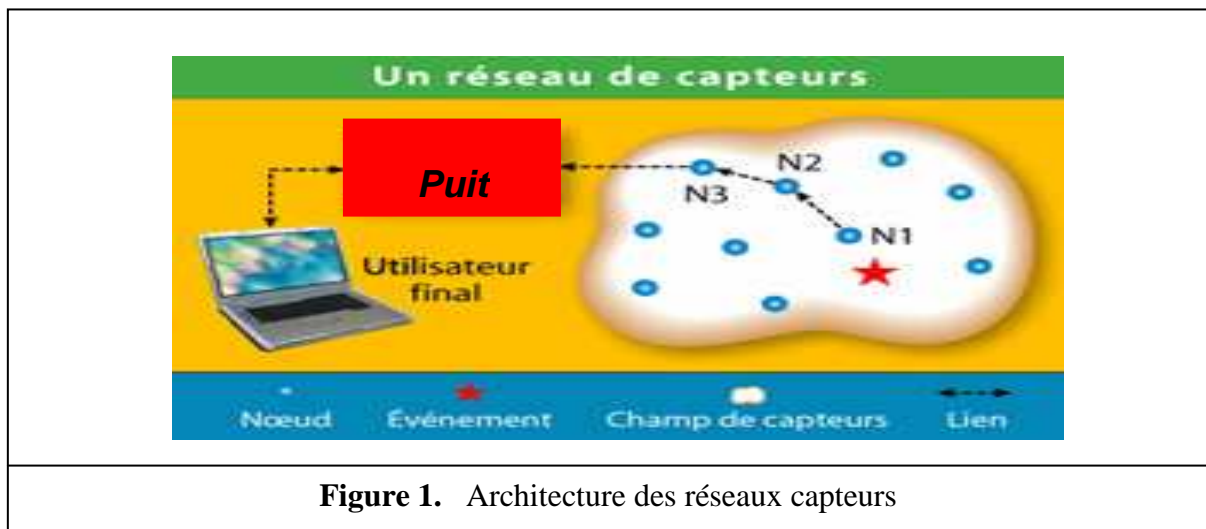
Premier chapitre

Premier chapitre : Le clustering des réseaux de capteurs et le problème d'énergie

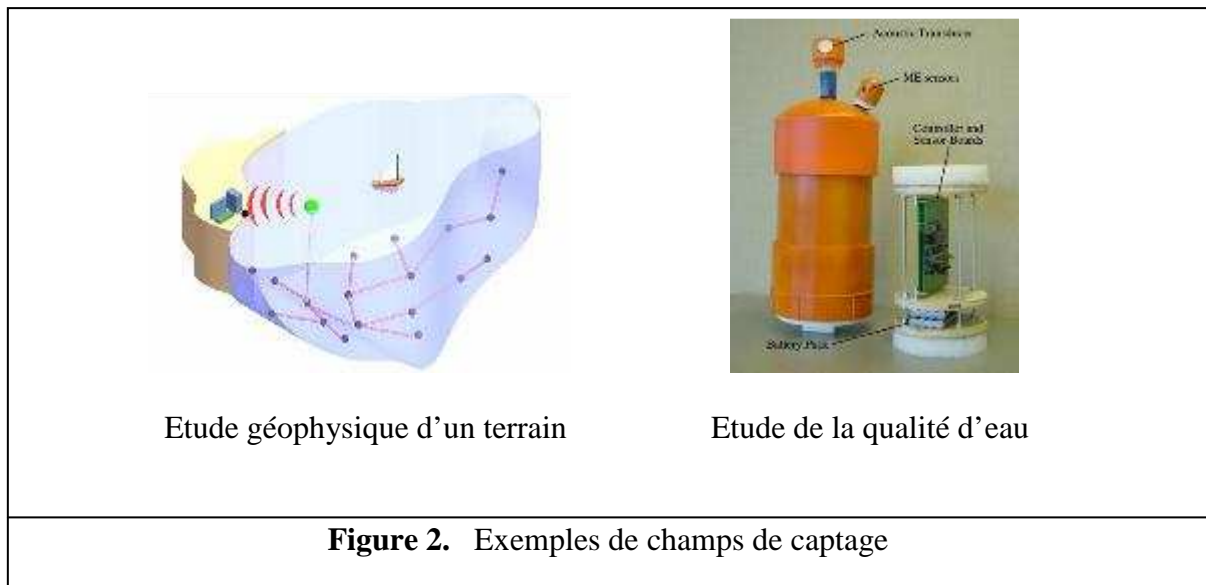
Le clustering des réseaux de capteurs et le problème d'énergie

I Architecture et terminologie

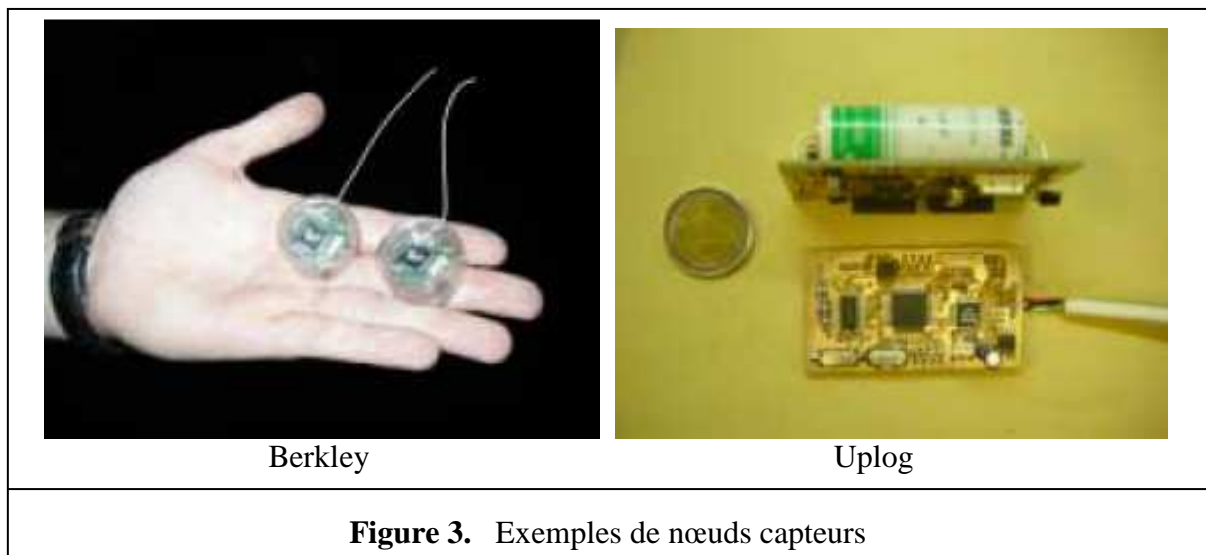
Un réseau de capteur est un réseau mobile dédié à l'étude d'un phénomène particulier. Il est constitué d'un ensemble de nœuds capteurs qui sont dispersés de manière dense et arbitraire sur le champ de surveillance. Les nœuds forment un réseau sans infrastructure et n'ont aucune information sur la topologie globale et même locale du réseau construit. Ils doivent graduellement établir l'infrastructure de communication durant une phase d'initialisation. Cette infrastructure doit leur permettre de répondre aux requêtes venant des sites distants, d'interagir avec l'environnement physique, réagir aux données captées et transmettre ces données via une communication multi-sauts [Tian 03].



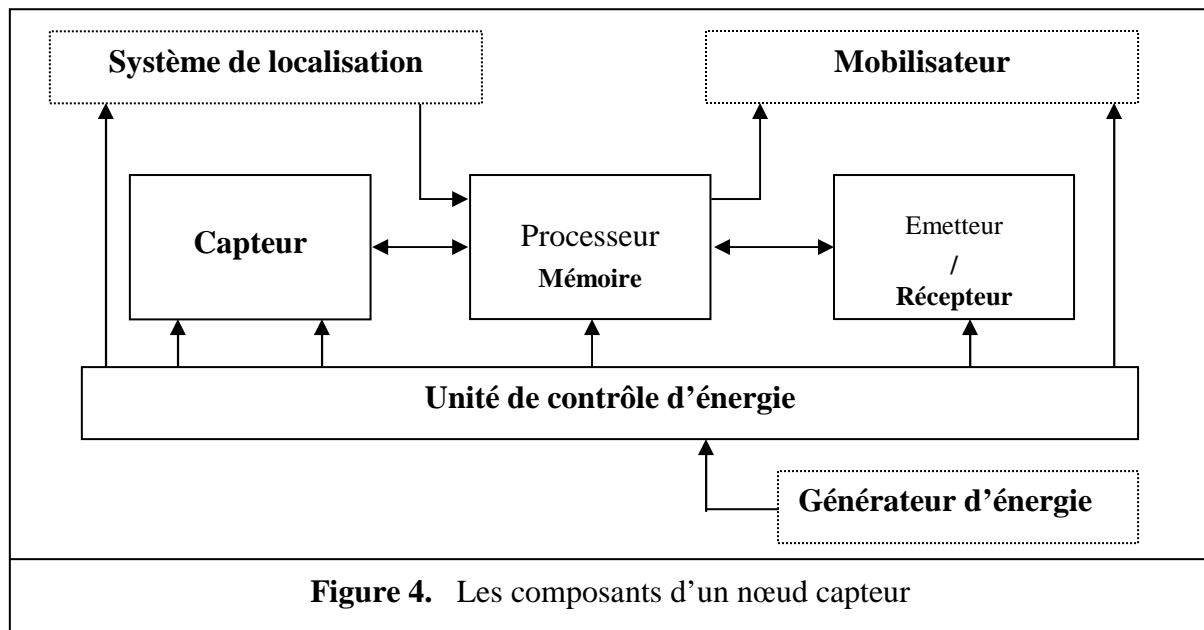
Champ de capteurs : c'est l'environnement sur lequel s'applique l'étude du phénomène. Ça peut être un corps humain pour le domaine sanitaire, les décombres dans une catastrophe naturelle ou un terrain pour une étude géophysique ...etc.



Nœuds capteurs : c'est des composants micro-capteurs dotés de dispositifs de mesure (captage), de traitement (calcul et mémoire), de communication (sans fil) et de contrôle d'énergie. Chacun des nœuds doit être capable de router les données au nœud puit.



En plus à ces quatre composants, il peut contenir également, suivant son domaine d'application, des éléments supplémentaires tels qu'un système de localisation, ou bien un système générateur d'énergie.



L'unité de captage englobe généralement deux sous-unités, le capteur lui-même en plus d'un convertisseur analogique-numérique qui transforme les signaux analogiques produits par les capteurs et basés sur le phénomène observé en signal digital, ce dernier est transmis par la suite à l'unité de traitement.

L'unité de traitement, généralement associée à une petite unité de stockage, exécute les procédures permettant au nœud de collaborer avec les autres nœuds du réseau pour enfin, donner le résultat de la tâche assignée au réseau.

La connexion du nœud au réseau est gérée par l'unité de transmission.

Enfin, l'unité de contrôle d'énergie constitue l'un des systèmes les plus importants dans un nœud capteur, celle-ci peut être représentée par un système de rechargement d'énergie telles que les cellules solaires.

Puits : il se charge de faire passer la requête depuis l'utilisateur vers le réseau, de collecter les réponses des capteurs, les fusionner et enfin envoyer le résultat à l'utilisateur. Le puit relie le réseau au monde extérieur.

Utilisateur final: c'est un nœud qui coordonne les tâches des nœuds capteurs et interroge le réseau pour d'éventuelles informations. Il fait passer sa requête au nœud puit via internet ou par satellite.

Evènement : les nœuds capteurs initialisent leurs opérations de captage soit par intervalles de temps, ou bien par déclenchement d'événements. Ces derniers peuvent être produits par des requêtes interrogatrices générées par l'utilisateur ou alors déclenchés localement, par exemple une valeur limite du phénomène étudié est atteinte.

Lien : c'est un chemin entre une paire de nœuds, il sera utilisé pour le routage de données.

Différents des réseaux Ad hoc traditionnels, les réseaux de capteurs engendrent de nouvelles limitations pour la conception des protocoles de communication. Par exemple les protocoles de routage doivent, en plus de leurs fonctions classiques, participer à la synthèse et l'agrégation des données retournées aux utilisateurs, tout en considérant d'autres facteurs tels que les limitations matérielles des micro-capteurs, la mobilité et la consommation d'énergie [Khelladi 04].

Le tableau suivant illustre les différences majeures entre les réseaux de capteurs et les réseaux Ad hoc [Fleury et al. 04] :

	Capteurs	Ad hoc
Domaine d'utilisation	Objectif ciblé	Générique / communication
Collaboration des noeuds	Nœuds collaborent pour remplir un objectif	Chaque nœud a son propre objectif
Flot de données	Flot de données : many-to-one ou one-to-many	Flot de données : any-to-any
Présence d'ID	Très grand nombre de nœuds n'ayant pas tous une ID	Notion d'ID
Propriétés physiques.	Energie et un facteur déterminant	Débit est majeur
Type des communications	Utilisation du broadcast	Communication point-à-point

Tableau 1. Différences entre les réseaux de capteurs et les réseaux Ad hoc

II Pile protocolaire

La pile protocolaire utilisée par les nœuds puit et capteurs prend en charge le problème de consommation d'énergie, intègre le traitement des données transmises dans les protocoles de routage et facilite le travail coopératif entre les capteurs. La figure suivante illustre l'architecture de cette pile protocolaire [Akyildiz et al. 02].

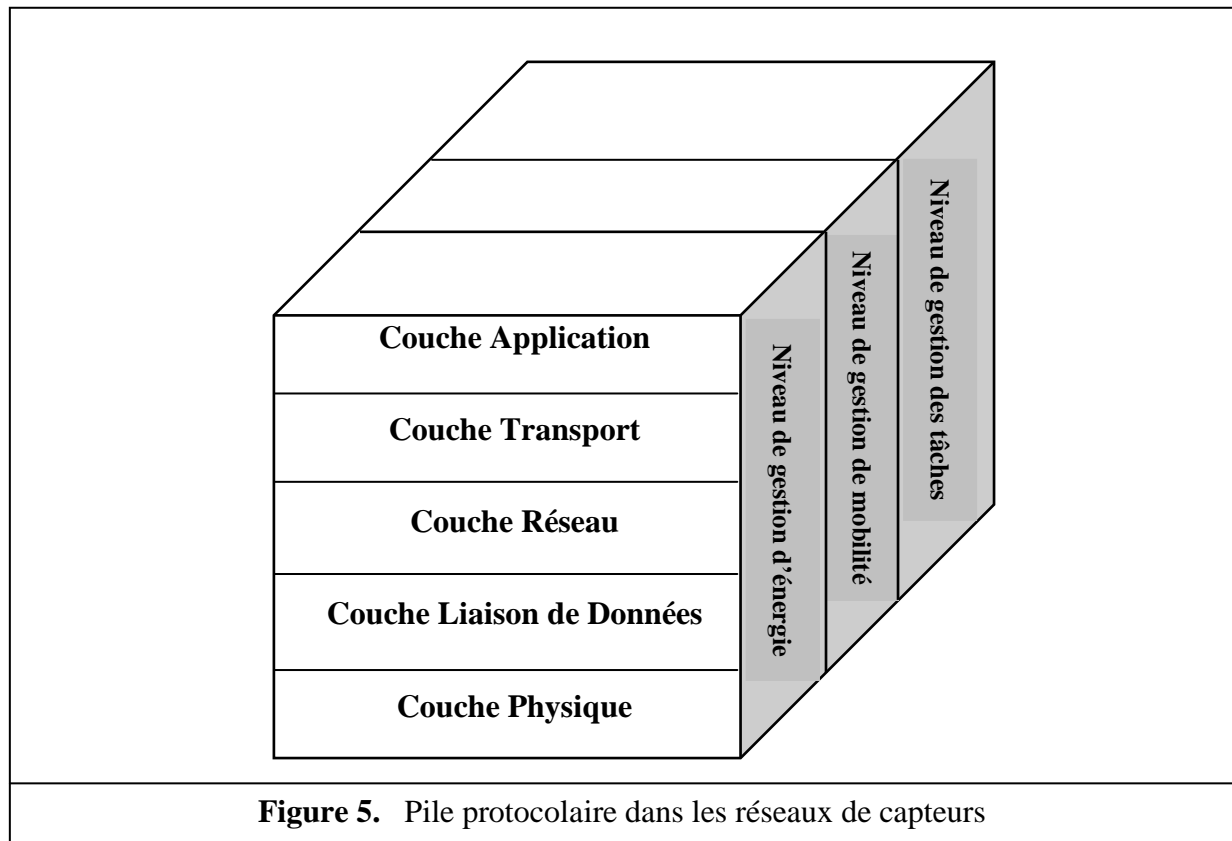


Figure 5. Pile protocolaire dans les réseaux de capteurs

II.I Le niveau gestion d'énergie

Il encapsule des fonctions qui consistent à gérer l'énergie consommée par les capteurs. Par exemple, un capteur peut éteindre son interface de réception dès qu'il reçoit un message d'un nœud voisin afin d'éviter la réception des messages dupliqués. De plus quand un nœud possède un niveau d'énergie faible, il peut diffuser un message aux autres capteurs pour ne pas participer aux tâches de routage et se concentrer sur les fonctionnalités de captage.

II.II Le niveau gestion de mobilité

Ce niveau permet de garder de manière continue un chemin depuis les nœuds capteurs vers l'utilisateur final et ce en enregistrant leurs mouvements, ce qui permet aussi d'avoir une image fraîche sur les nœuds voisins et ainsi équilibrer l'exécution des tâches et la consommation d'énergie.

II.III Le niveau gestion des tâches

Dans un réseau sensor et lors d'une opération de captage, les nœuds ne doivent pas forcément travailler avec le même rythme, ceci dépend essentiellement de la nature du capteur, de son niveau d'énergie et de la région dans laquelle il a été déployé. Le niveau de gestion des tâches assure l'équilibrage et la distribution des tâches sur les différents nœuds du réseau afin d'assurer un travail coopératif et efficace en matière de consommation d'énergie et par conséquent, prolonger la durée de vie du réseau [Akyildiz et al. 02].

II.IV La couche application

La conception de protocoles agissant dans la couche application reste largement inexploitée. Tout réseau de capteurs basé sur le schéma de couches protocolaires décrit précédemment nécessite l'utilisation des trois protocoles d'application suivants [Khelladi 04] :

- **SMP** (Sensor Management Protocol) : pour définir les règles d'agrégation et du nommage basé attributs, les données liées aux algorithmes de localisation, la synchronisation entre les nœuds, la gestion des déplacements, configuration/reconfiguration du réseau, la mise en marche/arrêt des nœuds et L'authentification, la distribution des clefs et la sécurité dans la transmission de données.
- **TADAP** (Task Assignment and Data Advertisement Protocol) : il consiste à définir l'approche utilisée afin de récupérer les données captées par les nœuds, à savoir :
 - Dissémination des requêtes « querying » : l'utilisateur interroge les nœuds capteurs.
 - Assignment de tâches « Task Assignment » : les nœuds capteurs envoient périodiquement les données captées aux utilisateurs.

- **SQDDP** (Sensor Query and Data Dissemination Protocol) : pour l'interrogation du réseau. Il fournit un moyen pour créer et répondre aux requêtes, ainsi que pour collecter les informations provenant des nœuds capteurs.

II.V La couche transport

Une variante de TCP qui est « TCP Splitting » peut être nécessaire afin de permettre aux réseaux sensor d'interagir avec les réseaux externes [Bakre et al.95].

Dans cette variante, le protocole TCP n'est utilisé qu'entre le réseau externe et le nœud puit, ceci est dû au fait que TCP se base sur la notion d'adresse unique qui est l'IP, chose qui n'est pas le cas pour les réseaux de capteurs. Quant au sein du réseau sensor, la communication peut être faite purement via UDP.

Plusieurs défis sont associés au développement des protocoles de transport fiables dans les réseaux de capteurs. En effet, dans le cas de réassignation de tâches (reprogrammation des tâches des nœuds) par exemple, cette application peut exiger plusieurs contraintes tel que l'adressage d'un certain groupe de capteurs, le chargement de nouveaux programmes, et le basculement vers les nouvelles tâches d'une manière contrôlée. Ces défis peuvent s'accroître encore si le nombre de nœuds capteurs augmente dans le réseau jusqu'à des milliers de nœuds [Khelladi 04].

Un nouveau protocole de transport PFSQ (Pump Slowly, Fetch Quickly) a été proposé. Il prend en charge ces nouveaux types d'applications dans les réseaux de capteurs qui nécessitent une transmission fiable de données.

II.VI La couche réseau

Comme nous l'avons déjà mentionné précédemment, les protocoles proposés dans les réseaux Ad hoc ne peuvent être appliqués aux réseaux de capteurs vu leurs spécificités et contraintes, surtout en terme d'énergie, caractérisant les nœuds capteurs.

De même, les techniques liées au routage doivent surmonter les défis introduits par les réseaux sensor notamment l'absence d'un adressage global (ID unique des nœuds) et la priorité absolue donnée à la quantité d'énergie consommée.

Dans les réseaux de capteurs, le mot routage est souvent remplacé par dissémination de données, pour faire ressortir l'absence d'adressage global, la dépendance aux rôles locaux des nœuds et l'utilisation du traitement spécifique à l'application [SILVA et al. 04].

II.VII La couche liaison de données

Cette couche est responsable du multiplexage des flux de données, le contrôle des accès au support de transmission, la détection des trames et le contrôle des erreurs. Elle garantit également des connexions point-à-point et point-à-multipoint fiables durant le processus de communication entre les nœuds capteurs.

II.VIII la couche physique

Elle est responsable de la sélection des fréquences, la génération des ondes porteuses, la détection du signal, la modulation et le cryptage des données.

La pile protocolaire utilisée pour les réseaux sensor a donné une très grande importance à la survivabilité maximale du réseau global et ce en optimisant la consommation d'énergie par les nœuds. Ceci est géré par le niveau de gestion d'énergie qui touche à plusieurs couches de la pile protocolaire entre autres, la couche réseaux tel que beaucoup de recherches ont été concentrées sur les protocoles de dissémination. La conception d'un protocole de routage efficace dans les réseaux sensor doit faire face à beaucoup de critères.

III Critères de conception d'un protocole de routage

Afin d'assurer une bonne connectivité du réseau et de prolonger sa durée de vie, la conception d'un protocole de routage est influencée par de nombreux facteurs qui doivent être surmontés avant qu'une communication efficace ne puisse être accomplie. Ces facteurs sont les suivants [Al-karabi et al. 03] :

- ❑ Déploiement des nœuds : les nœuds peuvent être déployés de manière déterministe ou aléatoire. Dans le premier cas, les capteurs sont placés manuellement et les chemins de routage des données sont prédéterminés. Dans le deuxième cas, les nœuds sont

éparpillés aléatoirement formant une infrastructure Ad hoc et les routes sont faites par plusieurs sauts.

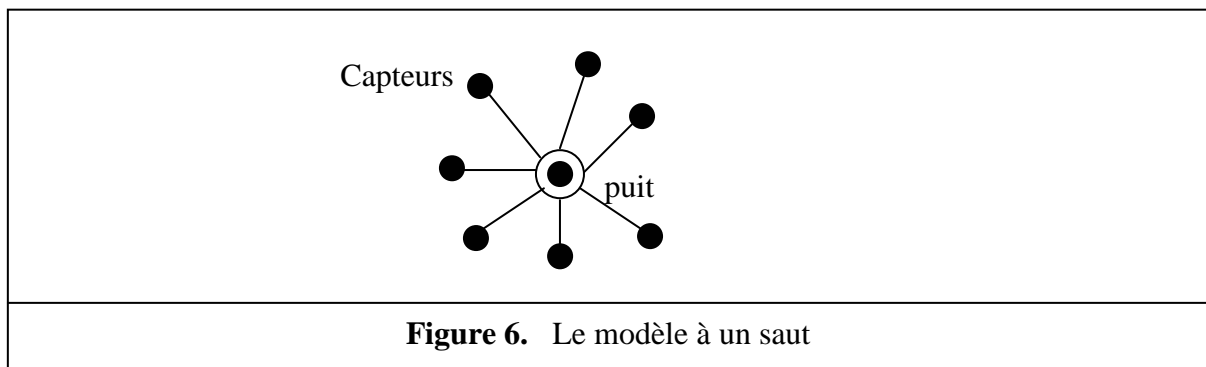
- ❑ Consommation d'énergie : la capacité des nœuds capteurs en terme d'énergie est très réduite, il faut alors que cette consommation soit hautement optimisée.
- ❑ Modèle de reportage de données : quatre modes de reportage de données sont à distinguer :
 - Continu : les nœuds transmettent les données captées à des intervalles de temps réguliers.
 - Événementiel : les nœuds réagissent après l'occurrence d'événements.
 - Suivant une requête : les nœuds réagissent après réception de requête.
 - Hybride : c'est une combinaison des modes précédents.
- ❑ Hétérogénéité des liens et des nœuds.
- ❑ Tolérance aux pannes.
- ❑ Scalabilité.
- ❑ Dynamicité des réseaux : déplacement et défection des nœuds, perturbation des liens de voisinage entre les nœuds.
- ❑ Médium de transmission.
- ❑ Connectivité.
- ❑ Couverture : chaque nœud du réseau a une certaine vue de l'environnement, cette vue peut couvrir une zone physique réduite et elle est limitée en exactitude.
- ❑ Agrégation des données : les paquets semblables peuvent être agrégés afin que le nombre de transmissions soit réduit.
- ❑ Qualité de service : parfois et pour des contraintes temporelles (applications temps réelles), les données doivent être fournies au cours d'une certaine période. Cependant, si l'énergie est épuisée, une dégradation de la qualité de service est impliquée. Afin de l'améliorer, il faut impérativement réduire la dissipation d'énergie.

IV Modes de dissémination

L'acheminement des paquets de données entre les nœuds d'un réseau peut suivre trois modèles différents, le premier est à base d'un saut, le second est basé sur deux ou plusieurs sauts et le troisième forme une hiérarchie arborescente.

IV.I Routage à base d'un saut :

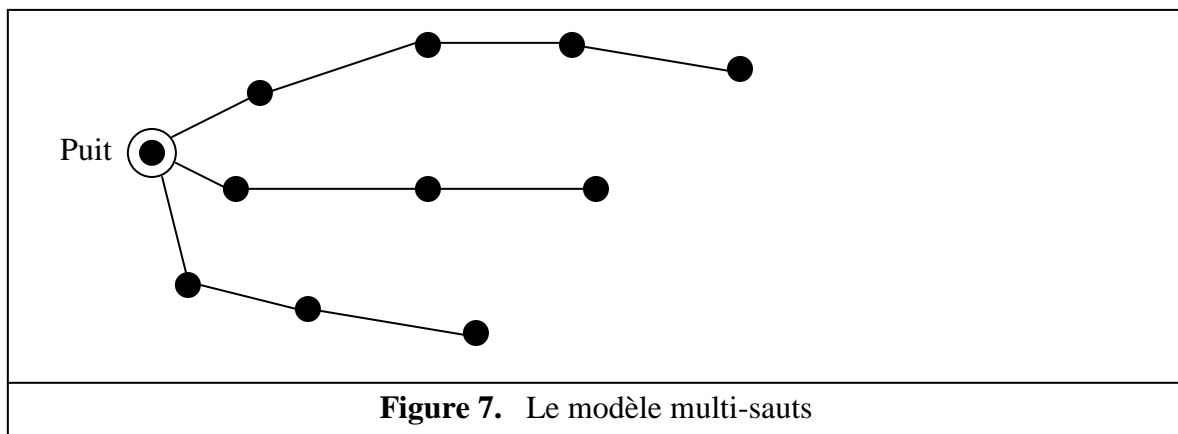
C'est le modèle le plus simple, il consiste à établir les communications entre les nœuds capteur et le puit de manière directe sans passer par un autre nœud.



Ceci est très coûteux en terme d'énergie, d'autant plus, cette solution est irréalisable à cause des interférences qu'elle peut impliquer.

IV.II Routage à base de plusieurs sauts :

Le nœud source de données envoie ses paquets à son voisinage proche qui les route à son tour de proche en proche jusqu'à atteindre le puit.

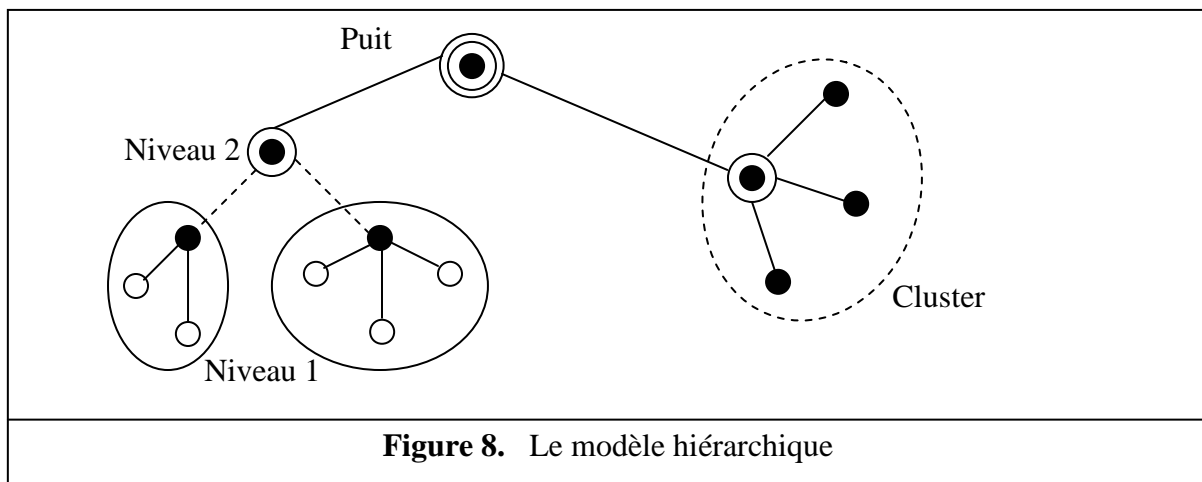


Son problème majeur réside dans le fait que le temps de dissémination devient très important lorsque le réseau est volumineux. De plus, le risque de collision (interférence entre les paquets) est toujours très important.

IV.III Routage hiérarchique :

Dans ce mode, le réseau global est décomposé en plusieurs sous groupes (cluster), chacun possédant une tête de groupe. Cette dernière prend la charge d'acheminer les paquets entre les sous groupes ou vers le puit. Le sous groupe peut être lui même décomposé en clusters.

Théoriquement, la latence (le temps écoulé entre la réception de l'interruption et son traitement) dans ce modèle est meilleure que celle des deux autres, mais l'entretien des sous groupes est une tâche difficile en tenant compte de la nature aléatoire des réseaux de capteurs [Ibriq 04].



Une technique qui permet de désigner la tête du cluster consiste à choisir aléatoirement un nœud capteur parmi plusieurs qui se sont proposés à l'élection. Cette opération se fait de manière périodique et les nœuds qui se proposent ne descendent pas au delà d'un seuil minimal d'énergie.

La plupart des protocoles de routage utilisent l'une des deux dernières techniques. Chacune ayant ses avantages et inconvénients en terme d'énergie, la technique d'hiérarchisation s'avère plus intéressante car un nombre très important de communications est éliminé en utilisant la tête de cluster. Cependant le nombre de messages nécessaires pour déléguer cette tête peut être très important, ce qui risque de consommer beaucoup d'énergie et épuiser les batteries des nœuds capteurs très rapidement.

Les protocoles de routage peuvent être regroupés sous plusieurs classifications possibles, le paragraphe suivant va illustrer les différentes classes existantes.

V Classes de protocoles de routage

Le but principal des protocoles de routage est d'établir des routes qui soient efficaces et correctes entre les nœuds du réseau sensor et le puit. Ceci doit être fait en minimisant les contrôles et la consommation d'énergie tout en tenant compte des contraintes liées aux réseaux de capteurs.

Plusieurs protocoles de routage ont été proposés dans la littérature, chacun apportant une solution par rapport à un critère donné. Ces derniers peuvent être regroupés sous quatre classes principales qui sont :

- Les protocoles centrés données.
- Les protocoles hiérarchiques.
- Les protocoles géographiques.
- Les protocoles orientés qualité de service.

V.I Les protocoles centrés données

Les protocoles orientés données sont basés sur l'utilisation des requêtes et l'emploi des systèmes d'identification de données, ce qui aide à éliminer les redondances de transmission.

En effet, l'identification s'applique sur les données et non pas sur les nœuds, cette idée permet de remédier au fait que les nœuds ne possèdent pas d'identification unique dans le réseau.

Les applications déployant les réseaux de capteurs s'intéressent plus aux données captées plutôt qu'aux capteurs eux même. Par exemple au lieu de formuler une requête demandant à un nœud « x » : quel est le nombre de véhicules captés par le nœud x, la requête peut alors avoir la forme : quelles sont les zones dans lesquelles le nombre de voitures excèdent 1000.

Cette approche découple les données des capteurs qui les produisent, ce qui mène à une conception plus robuste et pertinente.

V.II Les protocoles hiérarchiques

Ces protocoles visent à former des groupes et pour chaque groupe définir une tête de groupe capable d'agréger et de réduire les données dans un but de conservation d'énergie. Ils réduisent significativement le nombre de transmissions et ainsi la consommation d'énergie. Ils permettent aussi d'améliorer la scalabilité et de passer à un déploiement de réseaux intenses.

V.III Les protocoles géographiques

Ils se basent sur les informations de localisation des nœuds pour relayer les données aux régions désirées plutôt qu'à tout le réseau. L'information de localisation permet de calculer la distance entre une paire de nœuds capteurs de sorte que la consommation d'énergie puisse être déterminée, elle est utilisée aussi pour router les données à une région particulière.

V.IV Les protocoles orientés qualité de service

Les protocoles orientés qualité de service emploient des métriques de cette dernière (délai, énergie, bande passante, ...etc.) au moment de l'établissement des chemins entre les nœuds. Dans le cas d'un réseau de capteurs, le réseau doit faire un équilibre entre la consommation d'énergie et la qualité de données au moment de leur délivrance [Zeghilet 05].

La plupart des protocoles de routage peuvent être classés sous l'une de ces quatre catégories. Cependant nous pouvons trouver d'autres classifications citées dans la littérature basées sur d'autres critères tels que les protocoles à chemins multiples qui privilégient la tolérance aux pannes et ce en envoyant les paquets sur plusieurs chemins afin d'assurer leur réception même si certains nœuds situés sur les routes tombent en panne.

Dans la suite de ce document, notre travail se focalisera sur les protocoles de routage hiérarchiques. Ceci est dû à leur efficacité dans la réduction de la consommation d'énergie consommée et dans la qualité de l'agrégation des données.

VI Quelques protocoles de routage hiérarchique

Comme leur nom l'indique, les protocoles de routage hiérarchiques adoptent des chemins structurés sous forme hiérarchique (arborescente) afin de router les messages. Ces protocoles tentent d'assurer la scalabilité et de réduire la consommation d'énergie tels que le nombre de nœuds remonte à des milliers et la durée de vie du réseau de capteurs soit maximisée.

Les nœuds capteurs du réseau sont rassemblés en groupes, chaque groupe possédant une tête de groupe qui agrège et fusionne les données issues de son cluster et envoie les données agrégées au nœud puit. Par conséquent, le nombre de transmissions est réduit (les transmissions au puit seront faites seulement par les têtes plutôt que tous les nœuds), la charge sur le puit diminue et la surface couverte par le réseau ainsi que le nombre de capteurs peuvent être élargis de manière significative.

Cette tendance hiérarchique s'est imposée comme orientation très prometteuse pour la préservation d'énergie dans les réseaux de capteurs. Les protocoles proposés dans cette orientation ont prouvé des améliorations significatives suivant les métriques scalabilité et survivabilité du réseau global.

LEACH est l'un des premiers protocoles de routage hiérarchiques proposés, telle que l'idée de cette solution est à la base de beaucoup de protocoles hiérarchiques proposés ultérieurement. [Manjeshwar 01] [Lindsey 02], bien que certains d'entre eux aient été indépendamment développés [Subramanian 00].

VI.I Le protocole LEACH (Low Energy Adaptive Clustering Hierarchy)

Chandrakason et al. ont proposé dans le protocole LEACH de former des clusters de nœuds de capteurs basés sur la puissance du signal reçu et d'utiliser les têtes de clusters comme routeurs pour atteindre le puit. Ces dernières (têtes de clusters) collectent les données produites par les nœuds capteurs de leurs groupes, les agrègent par des procédures de fusion et transmettent les résultats de l'agrégation à la tête de cluster [Akyildiz et al. 02].

L'algorithme du protocole LEACH se déroule comme suit :

- Au début de chaque rond (période de temps pendant laquelle un nœud peut rester comme étant tête de groupe), chaque nœud vérifie s'il est capable d'être chef (tête) de groupe

pendant la période encours, dans le cas positif, il annonce sa décision à tous ses voisins en générant un nombre aléatoire entre 0 et 1. Si le nombre généré est inférieur à un certain seuil $T[n]$, le nœud est élu tête de groupe durant le rond courant. Le seuil $T[n]$ est donné par l'expression suivante :

$$T[n] = \begin{cases} \frac{p}{1 - p * (r \bmod 1/p)} & \text{Si } n \text{ dans } G \\ 0 & \text{Sinon} \end{cases} \quad [\text{Heinzelman et al. 00}]$$

Où « p » est le pourcentage du nombre de têtes de clusters par rapport au nombre total des nœuds. « r » est le nombre du rond courant. « G » est l'ensemble de nœuds n'ayant pas été chef de groupe durant les $1/p$ dernières périodes.

- Une fois la sélection des chefs des groupes est faite, ces derniers annoncent leurs décisions jointes de leurs identificateurs à tous les nœuds du réseau en utilisant le protocole MAC appelé CSMA [IEEE 802.3 Working Group].
- A la réception des annonces venant des chefs de groupes, tous les autres nœuds se joignent à l'un des chefs élus en se basant sur plusieurs paramètres prédéterminés tel que le rapport signal / bruit.
- Chaque nœud informe par la suite les têtes de clusters de l'identité du groupe auquel il a décidé de se joindre.

Dans le protocole LEACH, tous les nœuds sont considérés statiques, homogènes et ils sont prévus pour capter leur environnement et ont un taux de données fixe à transmettre, ce qui limite cet algorithme d'être déployé pour surveiller des objets mobiles.

Les auteurs présument que LEACH peut étendre la durée de vie du réseau par un facteur de 8 par rapport aux protocoles basés sur le routage multi-sauts [Heinzelman et al. 00].

Bien que LEACH puissent augmenter la durée de vie du réseau, il reste un certain nombre d'issus au sujet des hypothèses utilisées dans ce protocole qui sont :

- LEACH suppose que tous les nœuds peuvent transmettre avec une puissance permettant d'atteindre le puit, ce qui implique sa limitation dans les réseaux déployés dans de grandes régions.
- Aucune méthode permettant une distribution géographique uniforme des têtes de clusters à

travers le réseau n'a été définie. Par conséquent, des têtes de groupes peuvent être concentrées dans une région donnée, alors que des nœuds se trouvent isolés sans aucune tête de cluster.

- Les démarches de construction de cluster dynamiquement sont coûteuses en terme d'énergie.
- Dans l'étude liée à la simulation, 100 nœuds capteurs ont été considérés, alors qu'un réseau sensor peut facilement atteindre des milliers de nœuds. Ceci peut avoir un impact considérable sur les résultats de la simulation. De plus la possibilité de destruction des nœuds par des actions hostiles n'a pas été prise en charge dans les simulations [Rentala et al.01].
- Le protocole suppose que tous les nœuds ont la même quantité d'énergie dans chaque rond d'élection.

VI.II PEGASIS (Power-Efficient Gathering in Sensor Information Systems)

PEGASIS est une amélioration de LEACH, son idée clef est de former des chaînes entre les nœuds de sorte que chaque nœud envoie et reçoit les données seulement de son voisin le plus proche. Les données captées sont envoyées d'un nœud à un autre, après avoir été agrégées, jusqu'à atteindre le puit.

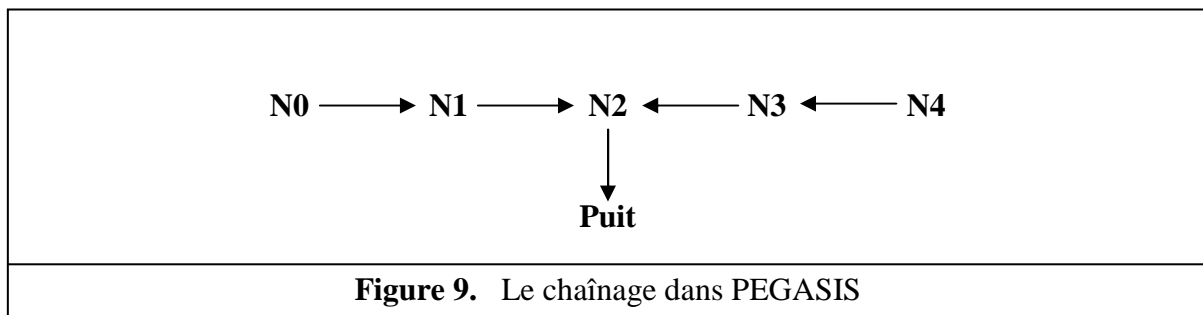


Figure 9. Le chaînage dans PEGASIS

N0 envoie ses données à N1, ce dernier fusionne ses données avec celles reçues par N0 et les transmet à N2. N2 passe le jeton à N4 qui par la suite envoie ses données à N3. Comme N1, N3 fusionne et envoie les résultats de cette fusion à N2. Enfin, N2 agrège l'ensemble des données reçues avec les siennes et les achemine vers le nœud puit [Lindsey 02].

La motivation principale pour ce protocole consiste à prolonger la durée de vie du réseau global et réduire la bande passante consommée en utilisant la collaboration locale entre les nœuds et en tolérant la défaillance des nœuds capteurs.

Bien que PEGASIS possède une structure hiérarchique tel que chaque nœud envoie les données à un nœud bien précis (voisin) pour atteindre le puit, le routage se ramène ainsi à un routage à base de sauts.

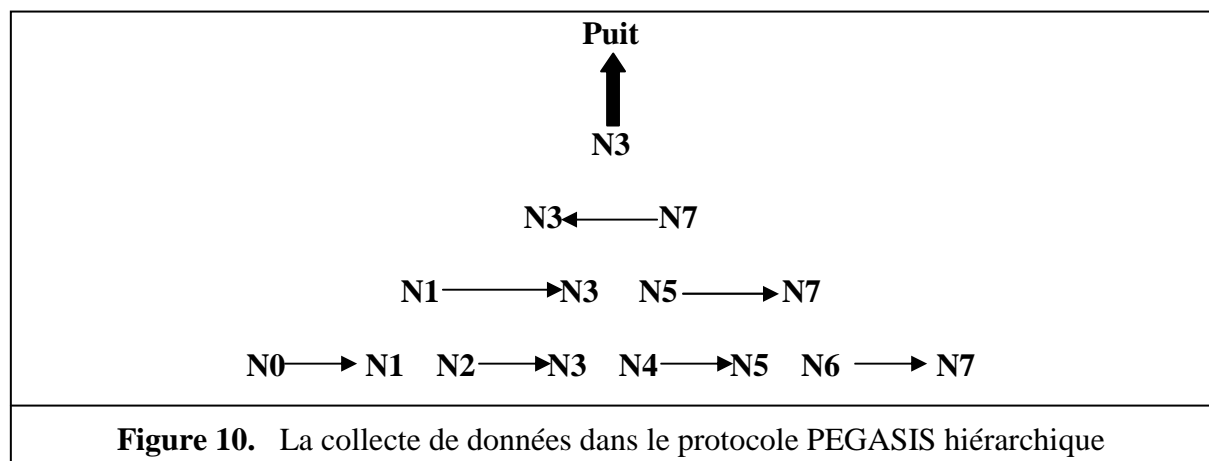
Les résultats de simulation ont montré que PEGASIS est meilleur que LEACH pour différentes tailles et topologies tel qu'il prolonge la durée de vie du réseau par un facteur de deux par rapport à LEACH [Lindsey 02].

Le gain est dû à l'élimination du coût relatif à la construction dynamique de clusters dans LEACH. Cependant, PEGASIS induit à un délai de transmission excessif pour un nœud distant sur la chaîne.

Une extension de PEGASIS a été proposée dans [Savvides 01] et qui consiste à diminuer le délai de transmission des paquets au puit, il s'agit de PEGASIS hiérarchique.

Dans PEGASIS hiérarchique, les messages sont transmis d'une manière simultanée et les nœuds constituent une chaîne sous forme d'un arbre. Chaque nœud se trouvant à un niveau particulier transmet les données au nœud du niveau supérieur de la hiérarchie.

Cette méthode assure une transmission parallèle des données et réduit les délais de manière significative et qui est de l'ordre de $O(\lg N)$ tel que N est le nombre de nœuds. La figure suivante explique le fonctionnement de PEGASIS hiérarchique :



Le nœud $N3$ est le responsable indiqué pour le rond.

- Tous les nœuds se trouvant dans des positions paires enverront les données à leurs voisins droits.
- Les nœuds récepteurs dans un niveau de l'hiérarchie montent au niveau supérieur ($N1$, $N3$,

N5, N7).

- De même, N1 et N5 passent leurs données respectivement à N3 et N7.
- N3 et N7 montent au niveau supérieur.
- A ce niveau, N3 est en position paire, donc N7 agrège ses données et les transmet à N3
- N3 combine ses données avec celles reçues de N7 et transmet enfin le message au puit.

Les résultats de simulation ont prouvé que ce protocole est plus performant que le PEGASIS simple, mais il exige toujours l'ajustement dynamique de la topologie (tout capteur doit connaître le statut de son voisin). Cet ajustement peut coûter des ressources de calcul significatives.

VI.III TEEN (Threshold sensitive Energy Efficient sensor Network protocol)

Le protocole TEEN est similaire à LEACH sauf que les nœuds n'ont pas un taux fixe pour transmettre, il est ainsi applicable aux réseaux réactifs (applications temps réel) où les nœuds n'ont besoin de transférer les données que si la valeur d'un attribut du phénomène surveillé atteint un seuil prédéfini [Manjeshwar 01].

Pour la formation des groupes, TEEN se comporte exactement de la même manière que LEACH, mais il adopte une approche différente pour la transmission des données. Il utilise deux paramètres appelés hard threshold (ht) et soft threshold (st). Le seuil hard est la valeur minimum possible d'un attribut pour déclencher un nœud de capteurs (capter et transmettre à la tête du cluster). Si un nœud capte une valeur inférieure ou égale au seuil hard, il transmet les données seulement quand le changement de valeur de cet attribut est supérieure ou égale au seuil soft.

Les valeurs de ht et st sont annoncées par la tête du cluster, elles permettent de réduire un nombre important de transmissions tel que les valeurs d'attributs ne sont transmises que lorsqu'elles appartiennent à une marge d'intérêt. Les seuils ht et st peuvent être ajustés afin de commander le nombre de transmissions de paquets.

Quoique TEEN offre une réduction importante du nombre de transmissions, il est cependant peu performant pour les applications dont les prélèvements des données doivent être effectués de manière périodique. De plus, si les valeurs d'attributs ne dépassent pas l'un des deux

seuils, aucune donnée des nœuds concernés ne sera transmise à l'utilisateur et le quanta de temps alloué à de tels nœuds sera inexploité.

Une autre limitation à ce protocole est due à la variation des taux de transmissions de données entre les différents nœuds suivant la grandeur de la variable captée, ce qui va mener à un épuisement rapide d'énergie au niveau des nœuds qui se trouvent dans une région ayant la valeur captée croissante d'une manière continue, ceci peut mener à une absence totale des nœuds capteurs dans de telles régions [Rentalala et al.01].

Une extension à TEEN a été proposée dans [Manjeshwar 02] qui offre la possibilité de faire des collectes périodiques de données, il s'agit de APTEEN (AdaPtive TEEN). Il a une architecture similaire à celle de TEEN, les têtes de clusters diffusent les attributs, les valeurs des seuils et le programme de transmission à tous les nœuds. Il utilise trois types de requêtes qui sont : historique pour analyser les valeurs de données déjà captées, instantanée pour avoir une vue sur l'état courant du réseau et de persistance pour surveiller un événement périodique [Zeghilet 05].

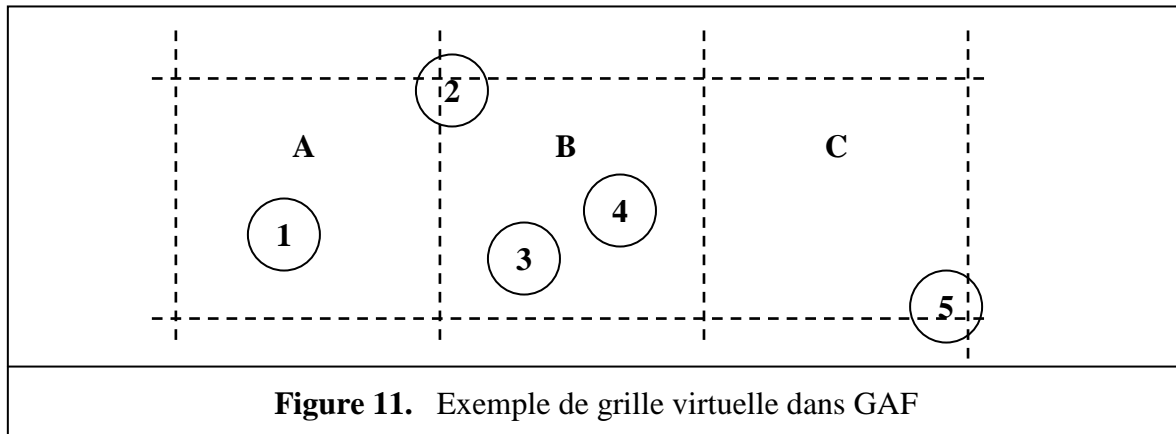
Les simulations ont montré que les performances de APTEEN sont entre celles de TEEN et de LEACH en terme de dissipation d'énergie et durée de vie du réseau de capteurs.

Ces deux protocoles souffrent de la complexité due à la formation de clusters dans différents niveaux, de mettre en application des fonctions à base de seuils et de traiter la description des requêtes en utilisant les attributs.

VI.IV Le protocole GAF (Geographic Adaptive Fidelity)

Ce protocole a été conçu pour les réseaux Ad hoc et a été appliqué aux réseaux de capteurs, les nœuds inutiles dans le réseau sont éteints et ainsi leur énergie sera conservée. L'algorithme forme une grille virtuelle pour une zone couverte. Connaissant son information de localisation, les nœuds s'associent avec un point dans la grille virtuelle [XU 01]

Les nœuds associés à un même point sur la grille sont considérés équivalents en terme de coût de routage et ainsi, certains nœuds peuvent être maintenus en état de sommeil (endormis). Les nœuds passent de l'état endormi à l'état actif alternativement de sorte que la charge soit équilibrée.



Pour chaque région, GAF garde toujours un nœud représentatif dans le mode actif afin de maintenir le réseau relié. Il est implémenté pour le cas statique (GAF-Bsic) et aussi pour prendre en charge la mobilité des nœuds (GAF mobility adaptation).

Ce protocole géographique peut être considéré hiérarchique tel que les régions correspondent aux clusters et le nœud représentatif à la tête du cluster mais sans aucune agrégation ou fusion.

Les résultats de simulation montrent que GAF fonctionne aussi bien dans les réseaux sensor que dans les réseaux Ad hoc tel qu'il augmente la durée de vie du réseau en économisant l'énergie.

VII Conclusion

Bien que les protocoles hiérarchiques aient considérablement diminué la consommation d'énergie, mais hélas, la phase de construction de clusters reste une tâche assez complexe et coûteuse. Il serait judicieux d'optimiser cette étape du point de vue nombre de messages de contrôle envoyés, le temps de construction du groupe et surtout la qualité du partitionnement du réseau obtenu.

Les cartes auto-organisatrices constituent une méthode de clustering non supervisée, efficace et optimale. Le protocole LEA2C est l'un des premiers protocoles qui ont utilisé les cartes SOM dans la procédure de clustering des réseaux sensor. Le chapitre prochain est consacré à la définition des cartes SOM (Self Organizing Maps) et de leur fondement théorique.

Deuxième chapitre

Deuxième chapitre : **Les cartes auto-organisatrices
de Kohonen (SOM)**

Les cartes auto-organisatrices de Kohonen (SOM)

I Introduction

Un réseau de neurones est un modèle de calcul dont la conception est schématiquement inspirée du fonctionnement de vrais neurones (humains ou non). Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type statistique, si bien qu'ils soient placés d'une part dans la famille des applications statistiques et d'autre part dans l'ensemble des méthodes de l'intelligence artificielle. En effet, les réseaux de neurones enrichissent les applications statistiques avec un ensemble de paradigmes permettant de générer de vastes espaces fonctionnels, souples et partiellement structurés. Ils enrichissent aussi les méthodes de l'intelligence artificielle en permettant de prendre des décisions s'appuyant aussi bien sur la perception que sur le raisonnement logique formel.

Les neurologues Warren Sturgis McCulloch et Walter Pitts ont mené les premiers travaux sur les réseaux de neurones à la suite de leur article fondateur : *What the frog's eye tells to the frog's brain*. Ils constituèrent un modèle simplifié de neurone biologique communément appelé neurone formel. Ils montrèrent théoriquement que des réseaux de neurones formels et simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes.

La fonction des réseaux de neurones formels à l'instar du modèle vivant est de résoudre des problèmes. À l'opposé des méthodes traditionnelles de résolution informatique, on ne doit pas construire un programme pas à pas en fonction de la compréhension de celui-ci. Les paramètres les plus importants de ce modèle sont les coefficients synaptiques. Ce sont eux qui construisent le modèle de résolution en fonction des informations données au réseau. Il faut donc trouver un mécanisme qui permette de les calculer à partir des grandeurs que l'on peut acquérir du problème. C'est le principe fondamental de l'apprentissage. Dans un modèle de réseaux de neurones formels, apprendre, c'est d'abord calculer les valeurs des coefficients synaptiques en fonction des exemples disponibles.

Le problème était donc de trouver une méthode qui permet d'adapter les paramètres synaptiques. Cette question était à l'origine de tous les travaux qui ont contribué à l'évolution des réseaux de neurones.

Les réseaux de neurones constituent une technique de traitement de données à l'aide de l'apprentissage. Ils permettent de tirer le maximum d'informations pertinentes de données qu'ils traitent à des fins de prévision, d'élaboration de modèles, de reconnaissance des formes, de classification ...etc

II Domaines d'application

Depuis leur apparition, les réseaux de neurones ont constitué une technique de modélisation, de traitement de données, de résolution de problèmes et d'optimisation des solutions.

C'est ainsi que les domaines d'application de ces réseaux sont devenus de plus en plus larges, tel que nous pouvons trouver dans la pratique un nombre très important d'applications qui sont à base des réseaux de neurones. Parmi ces applications, nous donnons à titre d'exemple :

- ❖ Application en reconnaissance des formes : la lecture automatique des codes postaux.
- ❖ Application en prévision : l'estimation de la probabilité de succès aux élections législatives.
- ❖ Application de fouille de données : filtrage de documents.
- ❖ Application en robotique : la modélisation de l'actionneur hydraulique d'un bras de robot.
- ❖ Application de classification : télédétection satellitaire
- ❖ Application de classification : carte topologique et recherche documentaire

Les domaines d'application des réseaux de neurones sont très variés à savoir : le datamining et le prévisionnel, la bio-ingénierie, la reconnaissance des formes, la robotique et commande de processus, la classification, l'auto-organisation ...etc.

Notre point d'intérêt se rapporte à ce dernier domaine qui est l'auto-organisation. La suite de ce chapitre est consacrée à l'étude des concepts et méthodes utilisées pour l'auto-organisation des réseaux de neurones.

III Définitions

Apprentissage : La notion d'apprentissage est une procédure cognitive qui permet à l'individu de réaliser à partir d'exemples et de manière autonome une tâche donnée. Prenons le cas d'un enfant, afin de lui apprendre à lire, on lui présente des exemples de lettres et de chiffres notés avec des écritures et fontes différentes. A la fin de l'apprentissage, on attend de l'enfant qu'il soit capable de lire non seulement tous les chiffres et lettres de son livre, mais également tous les chiffres et lettres qu'il est susceptible de voir. En d'autres termes, on attend de lui qu'il ait la capacité de généraliser à partir des exemples qui lui ont été présentés sans que ça soit nécessaire de lui fournir une description analytique et discursive de la forme des chiffres et des lettres [Dreyfus et al. 04].

Apprentissage numérique : il permet de définir un ensemble d'équations qui décrivent le processus observé et qui permettent de faire des prédictions concernant le comportement de celui-ci. Ceci est fait à l'aide d'une procédure numérique programmée et exécutée sur un ordinateur [Dreyfus et al. 04].

Un neurone est une fonction non linéaire, paramétrée et à valeurs bornées, dont les variables (les entrées du neurone) sont les observations (x_i) et les paramètres sont les référents (w_j) [McCulloch et al. 43], [Minsky et al. 69].

On appelle apprentissage des réseaux de neurones la procédure qui consiste à estimer les paramètres des neurones du réseau, afin que ce dernier remplisse au mieux la tâche qui lui est affectée. Deux types d'apprentissage sont à distinguer : l'apprentissage supervisé et l'apprentissage non supervisé [Dreyfus et al. 04] :

Apprentissage supervisé : Un apprentissage est dit supervisé lorsque le réseau est forcé à converger vers un état final précis, en même temps qu'on lui présente un motif. Nous connaissons en tous points ou en certains les valeurs que doit avoir la sortie du réseau en fonction des entrées correspondantes.

Apprentissage non supervisé : le réseau est laissé libre de converger vers n'importe quel état final optimal par rapport à un paramètre donné. C'est au réseau lui-même de découvrir les ressemblances entre les données étudiées. Les données sont représentées par des vecteurs de grande dimension, le but est

de chercher des regroupements selon des critères de ressemblance qui sont inconnus à priori. Ce type de tâches sont appelées « méthodes d'agrégation » et elles peuvent être effectuées par les réseaux de neurones à partir de données décrites par des vecteurs de grande dimension. Il s'agit de trouver une représentation de ces données dans un espace de faible dimension (typiquement 2 ou 3) tout en conservant les proximités ou ressemblances entre ces données.

Classer un ensemble d'objets, c'est attribuer à chacun une classe ou catégorie parmi plusieurs définies à l'avance. C'est typiquement l'un des cas d'utilisation des réseaux de neurones à apprentissage supervisé. Un algorithme qui réalise automatiquement la classification est appelé classifieur.

Par contre, l'auto-organisation permet d'obtenir des répartitions construites par le réseau lui-même, sans avoir recours à des paramètres prè-s-définis tel que le nombre de classes ...ect. Les réseaux de neurones à apprentissage non supervisé les plus étudiés et utilisés sont les cartes « auto-organisatrices » ou « carte de Kohonen ». C'est typiquement la modélisation du processus de clustering, tel que nous n'avons pas une structure du réseau prédéfinie à atteindre, mais plutôt la construire dans le temps selon l'état du réseau.

Dans la suite de ce document, nous nous intéressons spécialement à ces cartes afin de montrer leur utilisation et leur apport dans des tâches d'auto-organisation et la possibilité de leur exploitation dans les réseaux de capteurs.

IV Cartes auto-organisatrices SOM (Self Organizing Map)

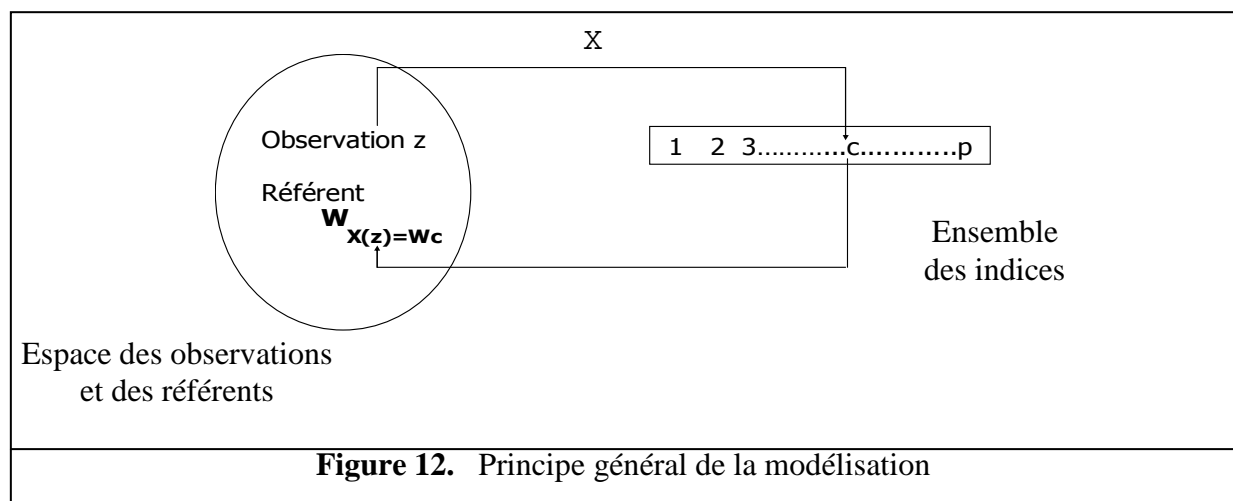
Les cartes topologiques auto-organisatrices font partie des modèles dits à « apprentissage non supervisé ». Ceci signifie qu'elles seront utilisées dans un but descriptif tel qu'il n'y a pas de structure précise à atteindre, ni de réponse souhaitée. Le but est alors de chercher à comprendre les structures des observations [Dreyfus et al. 04].

Les cartes topologiques ou cartes auto-organisatrices SOM (Self Organizing Map) ont été introduites dans un but de cerner de très grands ensembles de données, pour lesquelles les observations traitées pouvaient atteindre de grandes dimensions. Il s'agit de chercher à partir des données, à partitionner l'ensemble en groupements similaires. Ces derniers possèdent la particularité caractéristique d'avoir une structure de voisinage qui peut être matérialisée à l'aide d'un espace discret appelé « carte topologique ».

En effet, la classification automatique et les cartes auto-organisatrices sont des notions très proches étant donné que la plupart des méthodes de classification automatique cherchent à regrouper les données similaires. La notion d'ordre topologique constitue l'apport des réseaux de neurones à apprentissage non supervisé au domaine de la classification automatique qui est l'un des grands thèmes abordés en analyse des données. Une grande partie des applications qui ont été développées à l'aide des cartes SOM sont des classificateurs [Duda et al. 73], [Jain et al. 88], [Saporta 90].

IV.I Modélisation et terminologie

Ce paragraphe introduit les principes de base de la modélisation dans les réseaux de neurones ainsi que les notions utilisées dans le reste de ce document.



Soit D l'ensemble des observations réelles multidimensionnelles. L'espace D est de dimension n . $D \subset \mathbb{R}^n$. Chaque vecteur de D correspond à un codage particulier des individus issus d'une population donnée.

Nous disposons d'un sous ensemble d'observations A de D correspondant à N individus $A = \{z_i, i=1, \dots, N\}$ $A \subset D$

Dans un premier temps, le but est de réduire l'information contenue dans D en la résumant sous la forme d'un ensemble $W = \{W_c, c = 1, \dots, p\}$ de p vecteurs de D . Ces p vecteurs sont de dimensions n et sont appelés référents, ceci ne peut être fait qu'en définissant une fonction d'affectation qui est une application de D dans l'ensemble des indices $\{1, \dots, p\}$.

Cette fonction permet de réaliser une partition $P = \{P_1, \dots, P_c, \dots, P_p\}$ de D en p sous ensembles, $P_c = \{z \in D / X(z) = c\}$.

Ainsi, une observation z est associée à un indice c choisi parmi p à l'aide de la fonction X . cet indice permet de définir le référent W_c . Ce dernier est un représentant de l'ensemble P_c dont il résume les observations.

La connaissance de l'ensemble des vecteurs référents W et de la fonction d'affectation X , détermine une quantification vectorielle. Leur définition se fait par minimisation d'une fonction de coût. Cette dernière traduit les caractéristiques propres à la quantification que l'on veut obtenir. La minimisation est un processus itératif tel que chaque itération est constituée de deux étapes :

- ❖ Une étape de minimisation qui permet de déterminer les référents
- ❖ Une étape d'affectation qui redéfinit la fonction d'affectation

Dans le cas où ces deux étapes répondent à certains critères, ce formalisme assure que la fonction de coût décroît et converge vers un minimum local. Celui-ci dépend fortement des vecteurs référents choisis pour initialiser la procédure de minimisation.

Dans la littérature, il existe plusieurs méthodes permettant de réaliser cette minimisation, dont l'algorithme de Kohonen qui a été largement exploité dans différents domaines d'application.

Les cartes SOM proposées par Kohonen sont à base d'un algorithme d'auto-organisation qui projette l'espace des données D sur un espace discret de faible dimension (1, 2 ou 3), cet espace est appelé « carte » et noté C . Il est constitué d'un ensemble de neurones interconnectés suivant une structure de graphe non orientée. Cette structure induit une distance discrète δ sur C . Pour toute paire de neurones (c,r) de C , $\delta(c,r)$ est définie comme étant la longueur du plus court chemin entre c et r sur le graphe C . Cette distance discrète permet de définir la fonction de voisinage d'ordre d de c : $V_c(d)=\{r \in C, \delta(c,r) \leq d\}$ [Kohonen 01].

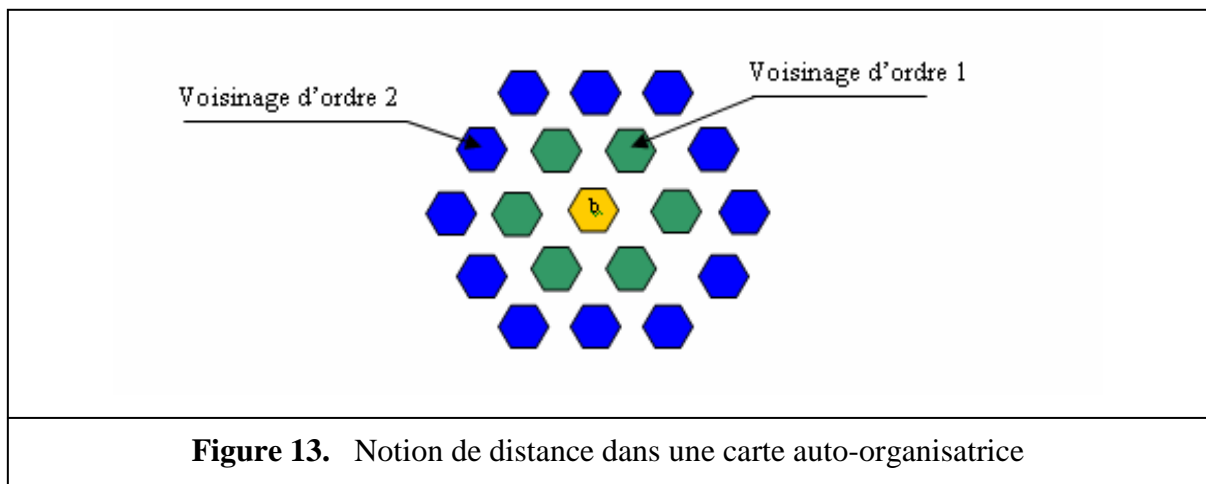


Figure 13. Notion de distance dans une carte auto-organisatrice

L'apprentissage effectué par les cartes SOM fait en sorte que les vecteurs référents captent au mieux la densité de probabilité sous-jacente aux observations. Ceci introduit une contrainte supplémentaire liée à la conservation de la topologie de la carte, tel que deux neurones c et r voisins sur la carte sont associés à deux vecteurs référents W_c et W_r proches par rapport à la distance euclidienne sur D .

L'algorithme des cartes SOM minimise une fonction de coût convenablement choisie. Cette fonction doit prendre en considération l'inertie interne de la partition de l'espace D et doit assurer la conservation de la topologie. Pour ce faire, les cartes SOM introduisent dans l'expression de la fonction d'inertie des termes spécifiques à partir de la carte. Cela est réalisé par l'intermédiaire de la distance définie sur la carte et de la notion de voisinage qui lui est attachée.

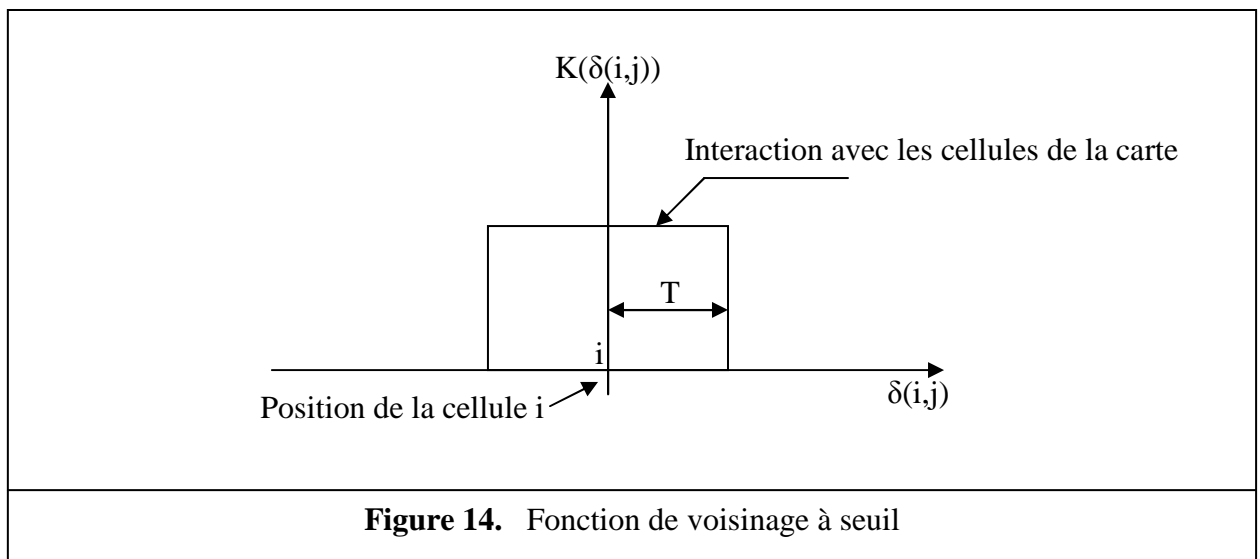
La notion de voisinage peut être exprimée à l'aide de fonctions noyaux positives et symétriques K ($\lim_{|x| \rightarrow \infty} K(x)=0$) [Dreyfus et al. 04].

Ces fonctions permettent d'introduire des zones d'influence autour de chaque neurone c . Les distances $\delta(c,r)$ qui lient le neurone c aux autres neurones (r) de la carte permettant de faire varier l'influence relative des différents neurones : cette importance est quantifiées par $K(\delta(c,r))$.

Afin de gérer la taille du voisinage, on utilise la famille des fonctions K^T paramétrées par T :

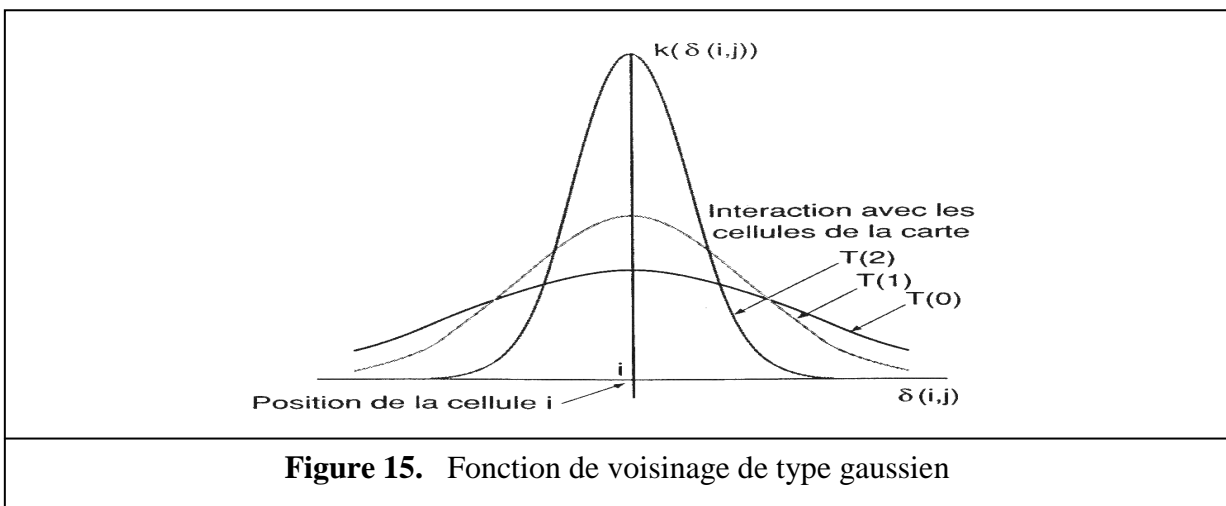
$$K^T(\delta) = K(\delta/T). \tag{1}$$

La figure suivante présente les fonctions noyaux les plus utilisées dans la pratique :



$$K(\delta) = \begin{cases} 1 & \text{si } \delta < 1 \\ 0 & \text{sinon} \end{cases} \quad \text{Ainsi, } K^T(\delta) = \begin{cases} 1 & \text{si } \delta < T \\ 0 & \text{sinon} \end{cases}$$

Les neurones du voisinage ont la même influence, en dehors, ils n'en ont aucune.



$$K(\delta) = \exp(-|\delta|) \text{ d'où } K^T(\delta) = \exp\left(\frac{-|\delta|}{T}\right)$$

$$K(\delta) = \exp(-|\delta|^2) \text{ d'où } K^T(\delta) = \exp\left(\frac{-\delta^2}{T^2}\right)$$

L'influence entre deux neurones, dépend de la distance entre eux.

Les algorithmes des cartes SOM minimisent une fonction de coût notée J_{som}^T et dont le minimum fournit une partition formée de sous-ensembles qui sont suffisamment compacts. La fonction J_{som}^T qui a été proposée pour le modèle le plus classique des cartes SOM a pour

$$\text{expression : } J_{som}^T(X,W) = \sum_{z_i \in A} \sum_{c \in C} K^T(\delta(c, \chi(z_i))) \|z_i - w_c\|^2 \quad (2)$$

Dans cette expression X représente une fonction d'affectation et W l'ensemble des p référents qui forment la carte. L'expression $X(z_i)$ représente le neurone particulier de la carte C qui est affecté à l'observation z_i et $\delta(c, X(z_i))$ représente la distance sur la carte entre un neurone quelconque et le neurone $X(z_i)$ affecté à l'observation z_i .

La distance euclidienne d^T d'une observation z_i à son référent $W_{X(z_i)}$ fait intervenir tous les neurones de la carte. Son expression est la suivante :

$$d^T(z_i, W_{X(z_i)}) = \sum_{c \in C} K^T(\delta(c, \chi(z_i))) \|z_i - w_c\|^2 \quad (3)$$

C'est une somme pondérée de la distance euclidienne de z à tous les vecteurs référents W_c du voisinage du neurone $X(z)$.

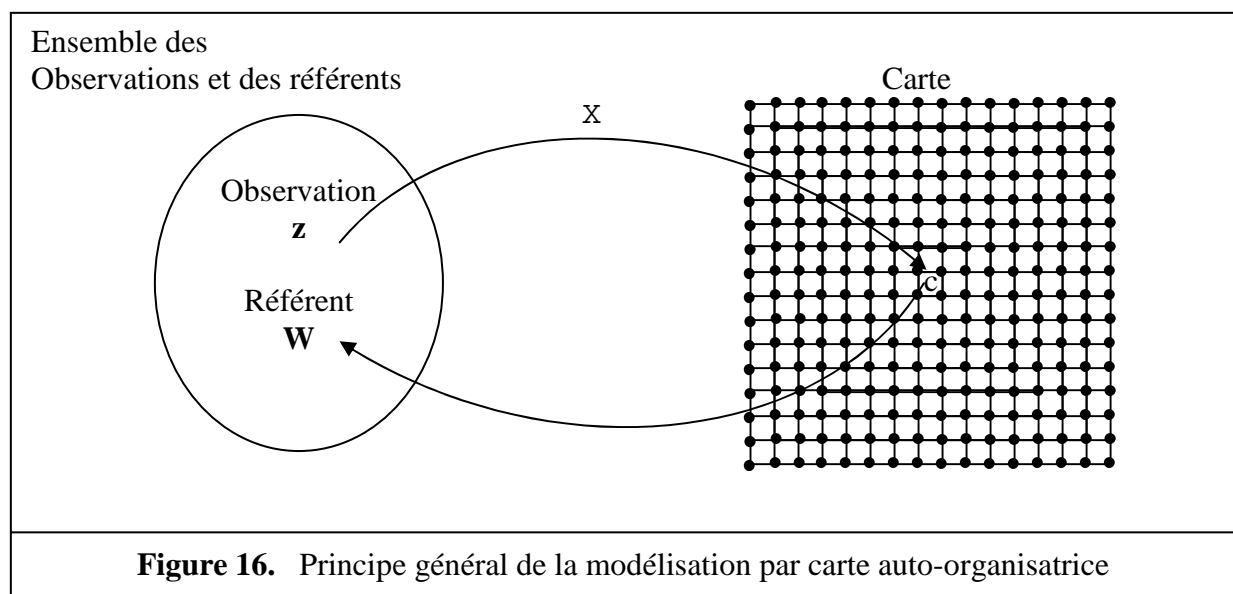


Figure 16. Principe général de la modélisation par carte auto-organisatrice

Les cartes SOM sont un cas particulier des réseaux de neurones. L'ensemble des indices est remplacé par les indices ordonnés de la carte.

Une observation z est associée à un indice c choisi dans la carte (parmi p neurones) à l'aide de la fonction $X(X(z_i=c))$. Cet indice permet de définir le référent W_c .

La minimisation de la fonction $J_{som}^T(X,W)$ peut se faire de différentes manières : adaptative, non adaptative ou alors probabiliste. Le paragraphe suivant abordera ces trois modes de minimisation [Dreyfus et al. 04].

IV.II Classes d'algorithmes d'optimisation de la fonction coût :

IV.II.1 Algorithmes d'optimisation non adaptative des cartes topologiques :

La minimisation de la fonction J_{som}^T pour une valeur T fixée est réalisée par itération successive, chacune se décomposant en deux phases. La première affecte l'ensemble des observations, la seconde minimise la valeur de la fonction coût associée à la partition.

- ❖ **Phase d'affectation :** Cette phase minimise la fonction $J_{som}^T(X,W)$ par rapport à la fonction d'affectation X . On suppose que l'ensemble des référents est constant et égal à la valeur calculée précédemment.

Les relations (2) et (3) montrent que l'affectation qui minimise J_{som}^T pour W fixé est celle qui est définie pour chaque observations z par :

$$X^T(z) = \arg \min_{r \in C} \sum_{c \in C} K_T(\delta(c,r)) \|z - w_c\|^2 = \arg \min_{r \in C} d^T(z, w_r) \quad (4)$$

Cette phase permet de définir une fonction d'affectation et une partition de l'ensemble des données D . Chaque observation z est affectée au référent le plus proche au sens de la distance pondérée d^T (3).

- ❖ **Phase de minimisation :** Elle permet de minimiser la quantité J_{som}^T par rapport à l'ensemble des référents W . Cette minimisation est effectuée en gardant la fonction d'affectation X fixée et égale à la fonction calculée durant la phase précédente. La fonction J_{som}^T étant convexe par rapport aux paramètres W , la minimisation est obtenue pour la valeur qui annule la dérivée, ce qui définit l'ensemble des nouveaux référents :

$$W_c^T = \frac{\sum_{r \in C} K(\delta(c, r)) Z_r}{\sum_{r \in C} K(\delta(c, r)) n_r} \quad (5)$$

Où $Z_r = \sum_{z_i \in A; X(z_i)=r} z_i$ représente la somme de toutes les observations de l'ensemble d'apprentissage qui ont été affectées au neurone r.

A. L'algorithme des nuées dynamiques :

Sur le plan algorithmique, la version « nuée dynamique » des cartes topologiques pour une valeur de T fixée se résume de la manière suivante :

Algorithme des nuées dynamiques des cartes topologiques : T fixé

1. **Phase d'initialisation : t = 0** Choisir les p référents initiaux (en général de manière aléatoire, la structure et la taille de la carte, le nombre d'itérations N_{iter}
2. **Etape itérative t** : l'ensemble des référents W^{t-1} de l'étape précédente est connu,
 - ❖ Phase d'affectation : mise à jour de la fonction d'affectation X associée à W^{t-1} . On affecte chaque observation z_i aux référents définis à partir de l'expression (4) ;
 - ❖ Phase de minimisation : Appliquer l'équation (5) afin de déterminer l'ensemble des nouveaux référents W^t
3. **Répéter** l'étape itérative jusqu'à ce que l'on atteigne N_{iter} itérations ou une stabilisation de J_{som}^T

L'expérience suivante permet d'illustrer le rôle du paramètre T dans la minimisation. Les observations sont équi-réparties entre quatre distributions normales qui se recouvrent partiellement deux à deux.

Les référents ont été initialisés d'une manière aléatoire au centre du nuage d'observations selon une gaussienne d'écart-type 0,01

Cette figure montre la carte obtenue pour quatre valeurs distinctes de T : 10, 5, 3 et 1.

Pour une grande valeur de T , l'ensemble des référents liés à la carte se regroupent d'une manière très dense au centre de gravité du nuage de points. Pour une petite valeur de T , les relations de voisinage interviennent moins et la carte se déplit à partir de la même initialisation.

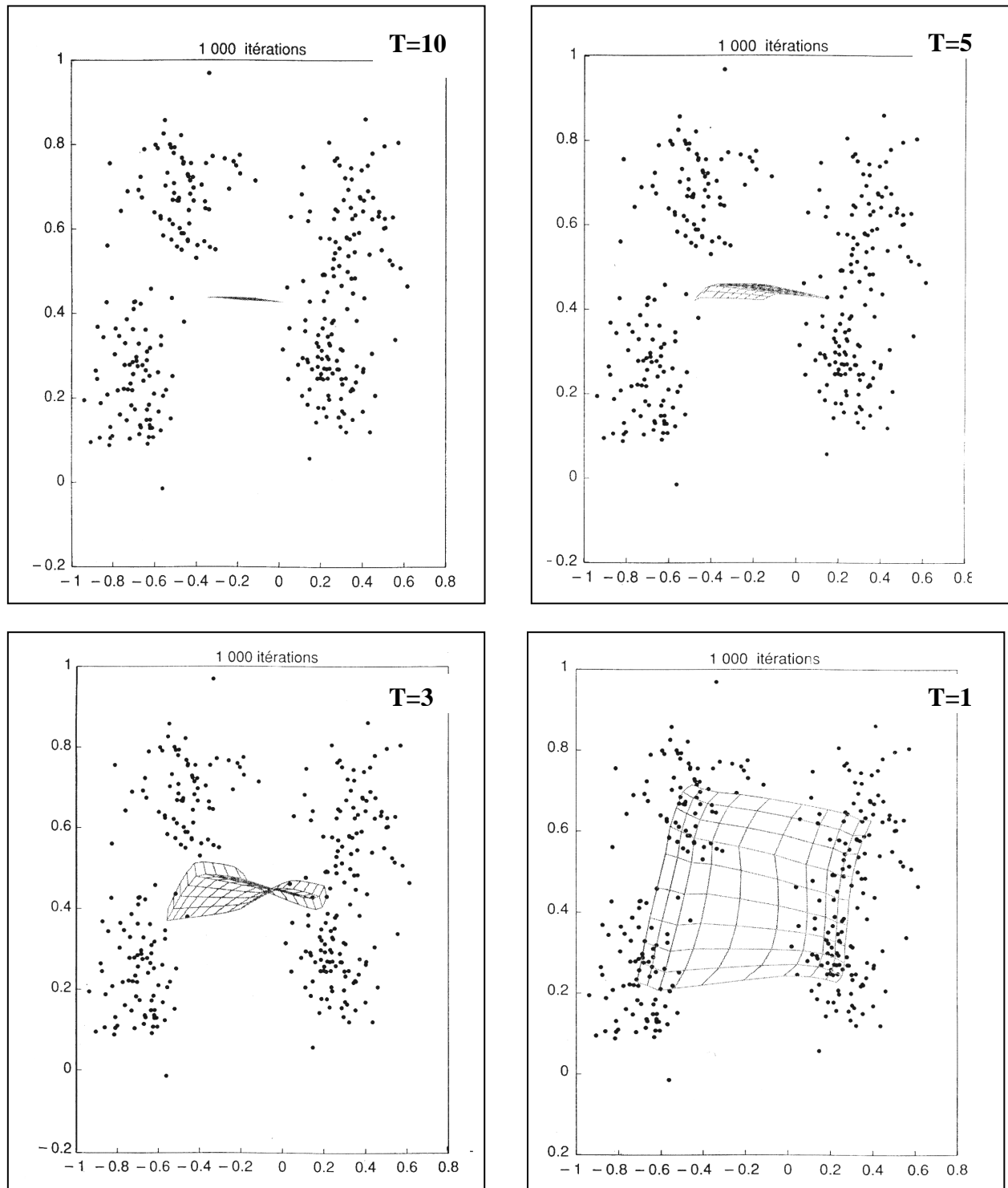


Figure 17. Déroulement de l'algorithme des nuées dynamiques à T fixé

La procédure qui vient d'être présentée permet donc d'atteindre pour une valeur fixée du paramètre T , un minimum local de la fonction de coût J_{som}^T par rapport à X et W . la procédure proposée par Kohonen répète cette minimisation un certain nombre de fois en faisant décroître la valeur de T .

B. Optimisation globale des cartes topologiques :

Dans cette approche, on passe donc progressivement par toutes les étapes présentées dans la figure précédente. Les référents étant initialisés de manière aléatoire, l'ordre apparaît au moment où la valeur de T est grande, la carte se déplie alors peu à peu de manière à recouvrir la distribution réelle des observations. Les propriétés du modèle que l'on obtient lors de la dernière itération, les qualités de la partition et de la quantification qui lui sont associées, ainsi que celle de l'ordre topologique, dépendent des différents paramètres utilisés durant la procédure de minimisation. Les paramètres déterminants de cette minimisation sont :

- ❖ L'intervalle de variation de T , la valeur initiale utilisée (T^{\max}) et la valeur finale atteinte (T^{\min})
- ❖ Le nombre de fois N_{iter} où l'étape itérative est effectuée
- ❖ La manière de décroissance de T dans l'intervalle $[T^{\min}, T^{\max}]$ [Dreyfus et al. 04]

La figure suivante met en évidence l'importance de la loi de décroissance sur la carte obtenue.

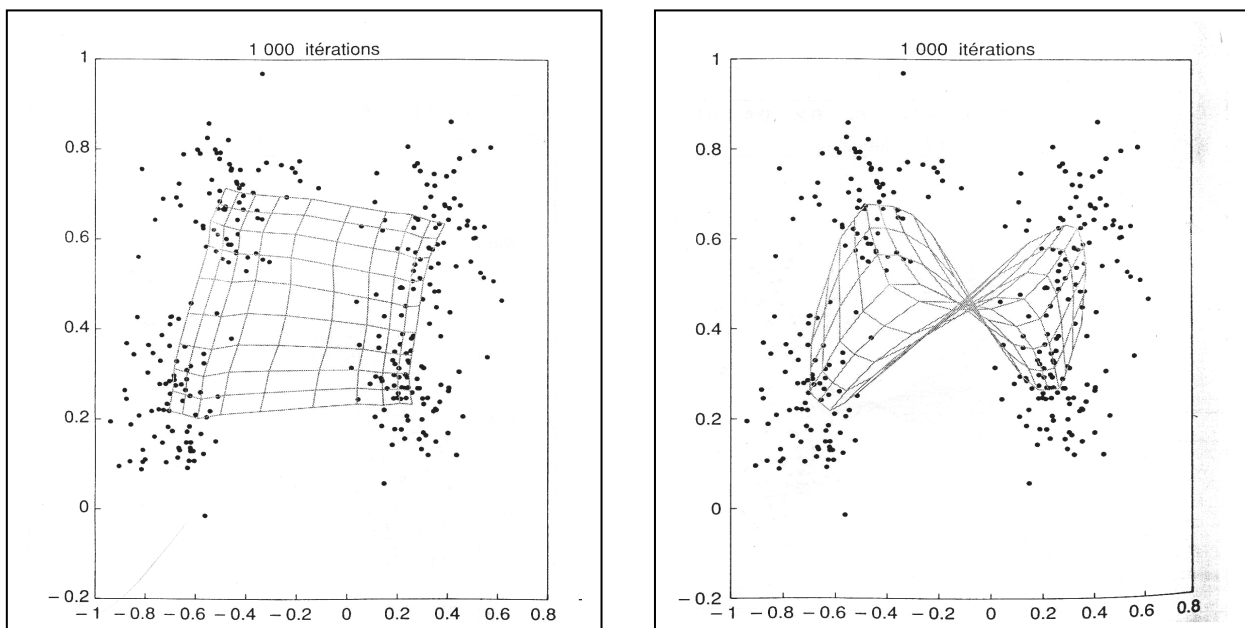


Figure 18. Représentation de l'ordre induit par la carte topologique pour deux décroissances différentes

On observe que si la décroissance est rapide, l'ordre se forme mal et les relations de voisinage ne représente pas l'ordre des sous ensembles. L'ordre topologique est très sensible à l'ensemble des paramètres qui interviennent dans l'algorithme.

Optimisation globale des cartes topologiques

1. **Phase d'initialisation** : Effectuer l'algorithme nuées dynamiques des cartes auto-organisatrices pour la valeur $T=T_{\max}$, $t=0$
2. **Etape itérative t** : l'ensemble des référents W^{t-1} de l'étape précédente est connu, calculer la nouvelle valeur de T en appliquant la formule :

$$T = T_{\max} * \left(\frac{T_{\min}}{T_{\max}} \right)^{\frac{t}{N_{iter}-1}} \tag{6}$$

Pour cette valeur du paramètre T, effectuer les deux phases suivantes :

- ❖ Phase d'affectation : mise à jour de la fonction d'affectation X associée à W^{t-1} . On affecte chaque observation z_i aux référents définis à partir de l'expression (4) ;
- ❖ Phase de minimisation : Appliquer l'équation (5) afin de déterminer l'ensemble des nouveaux référents W^t

3. **Répéter** l'étape itérative jusqu'à ce que l'on atteigne $T=T_{\min}$

La formule (5) montre que les cartes auto-organisatrices utilisent la fonction voisinage $K^T(\delta)$ paramétrée par T pour introduire l'ordre topologique. Pour des grandes valeurs de T, une observation z_i permet de modifier un grand nombre de vecteurs référents. A l'opposé, pour des petites valeurs de T, $K^T(\delta(c,r))$ est négligeable si $c \neq r$: une observation intervient uniquement dans le calcul du référent w_c qui lui est le plus proche. Les différentes valeurs de T utilisées pendant le déroulement de l'algorithme permettent aux vecteurs référents de la carte de se localiser. Plus précisément, la formule (5) montre que pour un paramètre T donné, le calcul du référent w_c dépend des observations de A qui appartiennent, d'une part, au sous-ensemble P_c et d'autre part, aux observations de P_r qui sont dans un voisinage significatif

$$r \in V_c^T = \{r / K^T(\delta(c,r)) \leq \alpha\}$$

Plus T est petit, moins le voisinage V_c^T contient de neurones et le nombre d'observations de A qui interviennent pour calculer w_c diminue. Pour des valeurs de T suffisamment petites, V_c^T se restreint au seul neurone c .

C. L'algorithme de Kohonen

Il est possible de définir une version stochastique de l'algorithme des cartes topologiques, il suffit de remarquer que lors de la phase de minimisation, il n'est pas obligatoire de trouver le minimum global de $J_{som}^T(X, W)$ pour X fixée, il suffit de faire décroître sa valeur. Il est donc possible de remplacer la relation (5) par une méthode de gradient simple. Ainsi, à l'itération t et pour un neurone c , on a :

$$w_c^T = w_c^{t-1} - \mu^t \frac{\partial J_{som}^T}{\partial w_c^{t-1}}$$

Où μ^t est le pas du gradient de l'itération t et :

$$\frac{\partial J_{som}^T}{\partial w_c} = 2 \sum_{z_i \in A} K^T(\delta(c, \chi(z_i)))(z_i - w_c) \quad (7)$$

Cette méthode non adaptative suppose que l'on dispose de toutes les observations de l'ensemble d'apprentissage A . La contribution d'une seule observation z_i à la correction de w_c est représentée par le terme de la somme $2K^T(\delta(c, \chi(z_i)))(z_i - w_c^{t-1})$. On peut même utiliser la méthode du gradient stochastique qui recalcule les référents chaque fois qu'une observation z_i est présentée. C'est cette version qui a été initialement présentée par Kohonen.

La différence par rapport à la version d'optimisation globale de l'algorithme présenté précédemment est que l'algorithme de Kohonen utilise une seule observation par itération et utilise comme fonction d'affectation celle utilisée dans l'algorithme des k-moyennes

$$\chi(z_i) = \arg \min_c \|z_i - w_c\|^2$$

A chaque présentation d'une observation z_i , les nouveaux référents sont alors calculés pour tous les neurones de la carte C en fonction du neurone sélectionné :

$$w_c^t = w_c^{t-1} - \mu^t K^T(\delta(c, \chi_t(z_i)))(w_c^{t-1} - z_i) \quad (8)$$

L'algorithme de Kohonen se résume donc de la manière suivante :

Algorithme de Kohonen

1. Phase d'initialisation :

- ❖ Choisir la structure et la taille de la carte et les p référents initiaux (en général d'une manière aléatoire)
- ❖ Fixer les valeurs de T_{\max} et T_{\min} et le nombre d'itérations N_{iter} , prendre $t=0$

2. Etape itérative t : L'ensemble des référents W^{t-1} de l'étape précédente étant connus :

- ❖ Choisir une observation z_i (en général de manière aléatoire)
- ❖ Calculer la nouvelle valeur de T en appliquant la formule :

$$T = T_{\max} * \left(\frac{T_{\min}}{T_{\max}} \right)^{\frac{t}{N_{iter}-1}} \quad (9)$$

Pour cette valeur du paramètre T , effectuer les deux phases suivantes :

- ❖ Phase d'affectation : On suppose W^{t-1} connu, on affecte l'observation z_i au neurone $X^t(z_i)$ défini à partir de la fonction d'affectation (3)
- ❖ Phase de minimisation : calcul de l'ensemble des nouveaux référents W^t . Les vecteurs référents sont modifiés selon la formule (8) en fonction de leur distance du neurone sélectionné à l'étape d'affectation

3. Répéter l'étape itérative en faisant décroître la valeur du paramètre T jusqu'à atteindre $t=N_{iter}$

IV.II.2 Algorithmes d'optimisation adaptative des cartes topologiques et cartes topologiques évolutives :

Les algorithmes qui viennent d'être présentés permettent de déterminer par apprentissage l'ensemble de tous les référents $W = \{ w_c ; c \in C \}$ d'une carte auto-organisatrice. On peut représenter l'ensemble constitué par la carte et les référents sous la forme d'un réseau de neurones constitué de deux couches :

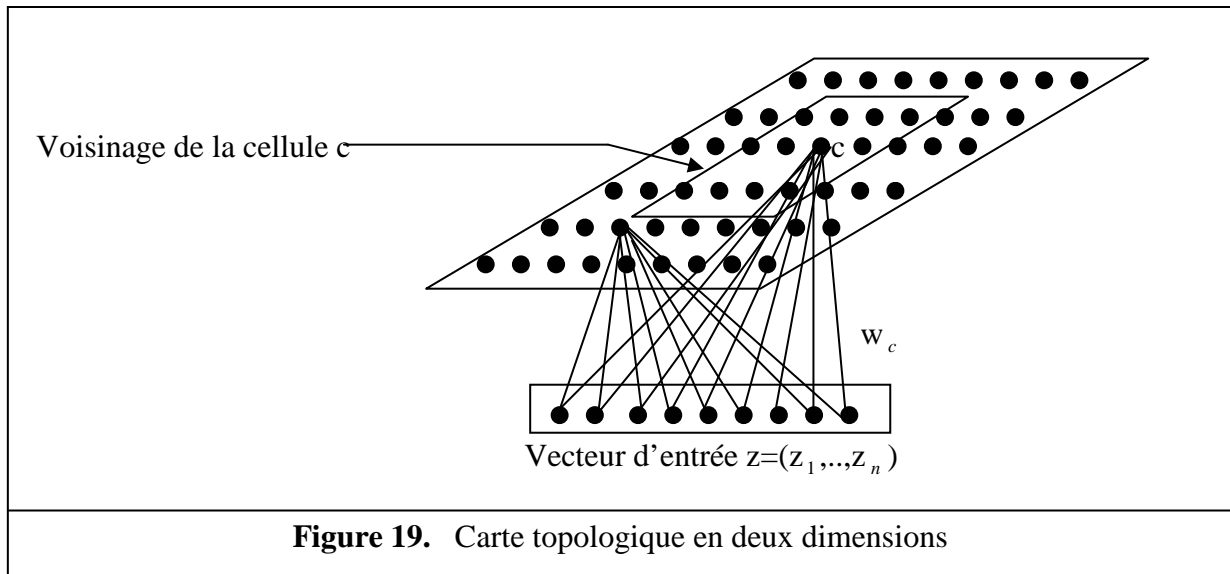


Figure 19. Carte topologique en deux dimensions

La couche d'entrée sert à la présentation des observations à classer. Les états de tous ses neurones sont forcés aux valeurs des observations. Cette couche contient donc exactement n neurones (n étant la dimension de l'espace des observations)

La couche d'adaptation est formée du treillis des neurones qui forment la carte. La structure du réseau employé peut soit être fixée a priori ou alors évoluer lors de l'apprentissage. Les neurones utilisés sont de simples neurones (distance). Chacun d'entre eux est entièrement connecté à la couche d'entrée. Le vecteur référent $w_c = (w_1, w_2, \dots, w_n)$ associé à un neurone c de la carte, n'est autre que le vecteur des connexions (ou vecteur des poids) qui arrive au neurone c . Puisque le réseau est totalement lié à la couche d'entrée, chaque neurone possède n connexions (poids). En réponse à une observation z , un neurone distance c de C détermine son état en calculant $\|z - w_c\|^2$

Afin de permettre que le processus d'auto-organisation s'effectue, les poids qui lient les deux couches du réseau sont adaptatifs : ils sont mis à jour à l'aide des différentes règles de

modification des référents qui ont été présentées. Dans ce réseau, les neurones de la carte calculent leur état (distance), en parallèle, à partir des mêmes informations fournies par l'observation qui figure en entrée.

La principale caractéristique du processus d'auto-organisation est de ne permettre une adaptation des paramètres du réseau que sur la région de la carte la plus active. L'algorithme le plus simple (celui de Kohonen) détermine ce centre d'activité comme étant le voisinage de la carte associé au neurone dont l'état ($\|z - w_c\|^2$) est le plus petit. C'est l'utilisation de ce voisinage qui introduit les contraintes topologiques dans la représentation finale. Comme cela a été signalé dans le paragraphe précédent, cela modélise de façon simplifiée un couplage latéral entre un neurone sélectionné et ses voisins dans la structure graphique de la carte. De cette façon, en fin d'apprentissage, les poids de chaque neurone convergent vers des valeurs telles qu'un neurone ne sera plus actif que pour un sous-ensemble bien déterminé d'observations de la base d'apprentissage. Un neurone c qui est représenté par son vecteur référent w_c peut être considéré comme une observation moyenne qui résume le sous-ensemble P_c des observations qui lui sont affectées. L'ensemble des neurones de la carte représentent donc une quantification vectorielle de l'ensemble D , obtenue par l'analyse de l'ensemble d'apprentissage A . La qualité de la quantification dépend donc de celle de l'ensemble d'apprentissage.

Dans les cartes auto-organisatrices, le problème majeur est de choisir une structure de carte bien adaptée au problème traité. Pour ce faire, il faut définir les critères qui permettent de juger de l'adéquation d'une carte à un problème donné. L'espace des données et la carte sont liés par deux applications : une application de C dans D qui permet d'affecter à un neurone c de la carte un référent w_c qui est une observation moyenne et une fonction d'affectation X qui permet d'associer à tout vecteur de D , un neurone c de la carte qui soit son représentant. Ces deux applications doivent vérifier la propriété de conservation de topologie :

- ❖ Deux neurones voisins dans l'espace de la carte doivent être représentés par deux référents voisins dans l'espace des données.
- ❖ Réciproquement, des données similaires doivent être affectées à des neurones voisins.

Dans les paragraphes précédents, les cartes ont été supposées être choisies à priori, indépendamment de la structure des données. Or cette façon de procéder présente des

inconvénients évidents, puisqu'un choix pareil peut être insuffisant pour capter la structure interne des données. Ainsi, la propriété de conservation topologique ne sera pas garantie.

En général, dans les applications, les observations n'occupent pas tout l'espace des données : elles sont réparties dans des régions dont les concentrations sont variables. Une manière de décrire cette répartition consiste à utiliser des fonctions densité de probabilité qui permettent de donner une description analytique de cette répartition. La carte doit modéliser cette densité pour bien répartir les référents afin de refléter au mieux la répartition des observations. Les référents doivent être concentrés dans les régions de forte densité et éviter les régions ayant une densité nulle.

Une manière de procéder est de considérer une carte surdimensionnée par rapport au problème traité et d'appliquer l'algorithme de Kohonen à cette carte. A la fin de l'apprentissage, on considère la partition $\{P_r; r \in C\}$ et l'on supprime de la carte les neurones r pour lesquels $P_r \cap A = \Phi$. L'élimination de ces neurones se justifie par le fait que les référents associés sont placés dans des régions de D où la densité de probabilité de données est nulle. On peut alors appliquer l'algorithme de Kohonen à la carte obtenue afin de réadapter la nouvelle carte aux données et itérer cette procédure autant que nécessaire.

La seconde technique consiste à définir la carte (nombre de neurones et structure graphique) lors de l'apprentissage, en même temps que la mise au point des référents. Il s'agit donc de construire la carte d'une manière évolutive, en permettant l'ajout de certains neurones et la suppression d'autres. Plusieurs méthodes ont été proposées dans la littérature, ces méthodes sont classées en deux catégories :

- ❖ La première catégorie fixe à priori la dimension k de la carte et la construit de manière évolutive par adjonction et suppression de neurones.
- ❖ La seconde laisse aux données elles-mêmes le choix de la dimension de la carte qui peut varier d'une région à une autre. L'algorithme « neural gas » [Oja et al. 99] construit le graphe en introduisant les connexions directement dans l'espace des données. Ainsi, chaque fois qu'une observation est présentée, on retient les deux référents les plus proches. S'ils sont déjà reliés par une connexion, alors cette dernière est réactivée, sinon elle est créée. L'algorithme supprime les connexions qui restent inactives après un nombre fixé d'itérations.

IV.II.3 Algorithmes d'optimisation probabiliste :

Il est possible de modifier l'algorithme des cartes topologiques afin d'en donner une version probabiliste. Ce modèle est appelé PRSOM, il associe à chaque neurone c de la carte une fonction densité normale f_c , la notion de voisinage permet d'introduire un ensemble de mélanges de gaussiennes. Chaque fonction densité est complètement définie par son vecteur moyen (vecteur référent) de dimension n : $w_c = (w_c^1, w_c^2, \dots, w_c^n)$ ainsi que par sa matrice de variance-covariance \sum_c qui est une matrice carrée de dimension n définie positive [Anouar et al. 97], [Gaul et al. 00].

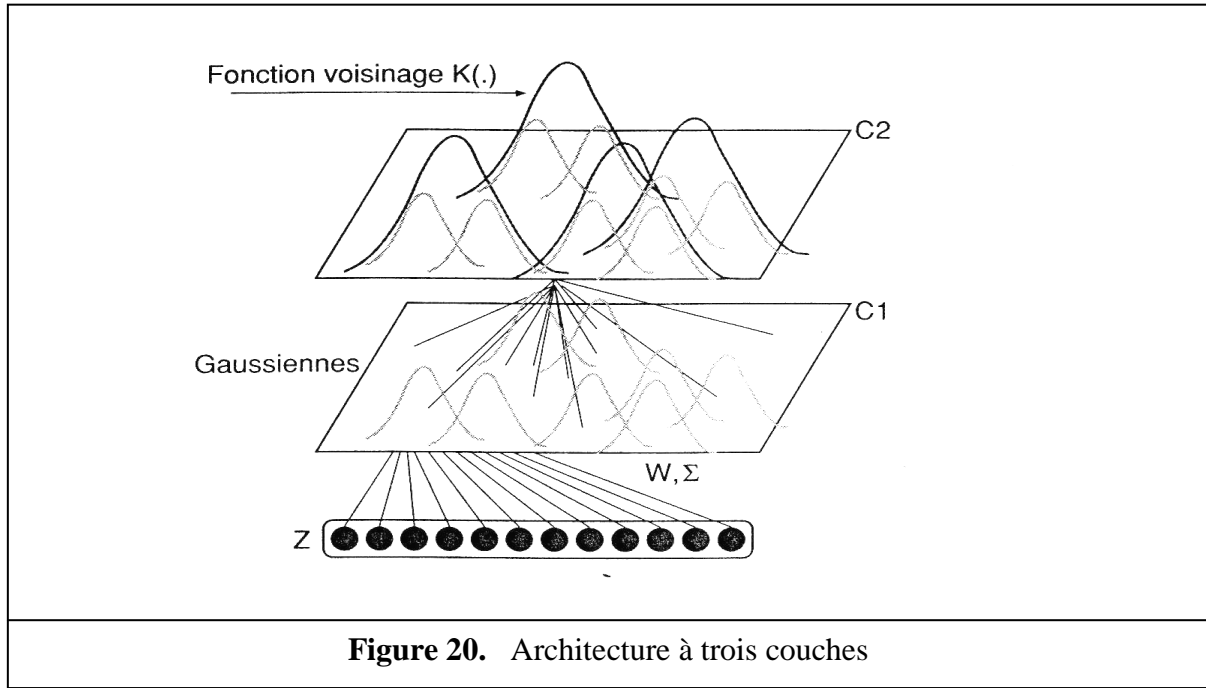
L'ensemble de ces fonctions ont pour expression :

$$f_c(z) = \frac{1}{(2\pi)^{\frac{n}{2}} \sigma_c^n} \exp\left(-\frac{\|z - w_c\|^2}{2\sigma_c^2}\right) \quad (10)$$

Ainsi dans le modèle PRSOM, on attribut à chaque neurone c de la carte, le vecteur moyen w_c et le nombre positif σ_c . Comme pour les cartes auto-organisatrices, l'ensemble D est partitionné par la famille $\{P_c / c \in C\}$. Le sous-ensemble P_c est décrit par la fonction densité f_c , w_c représente son référent et σ_c décrit la dispersion des observations de $P_c \cap A$ autour de w_c . Les deux ensembles $W = \{w_c, c \in C\}$ et $\sigma = \{\sigma_c, c \in C\}$ définissent complètement le modèle PRSOM. L'ensemble des valeurs de leurs éléments doivent être estimés durant la phase d'apprentissage en utilisant l'ensemble A .

Si la distribution sous-jacente aux données est un mélange de lois gaussiennes, le modèle PRSOM permet d'estimer les paramètres de ce mélange. Il est possible de donner une représentation neuronale du modèle. Le réseau correspondant à ce modèle possède une architecture à trois couches :

- ❖ La couche d'entrée sert à la présentation des observations à classer.
- ❖ La carte C est dupliquée en deux cartes semblables C_1 et C_2 qui ont la même topologie que la carte C présentée pour le modèle des cartes auto-organisatrices. Sur ces deux couches, on notera par c_1 (ou c_2) un neurone de la couche (C_1 ou C_2).



Le modèle PRSOM suppose qu'un phénomène de propagation probabiliste, dans les deux sens, se réalise à travers les trois couches du réseau. Ainsi, dans le sens qui va de la carte vers l'espace des données, cette propagation est décrite par les probabilités: $p(c_1|c_2)$ et $p(z|c_1, c_2)$. On suppose que ce modèle vérifie la propriété de Markov : $p(z|c_1, c_2) = p(z|c_1)$. Il devient possible d'exprimer la probabilité de chaque observation z sous la forme [Luttrell 94] :

$$p(z) = \sum_{c_2} p(c_2) p_{c_2}(z) \quad (11)$$

$$\text{Avec } p_{c_2}(z) = \sum_{c_1} p(c_1|c_2) p(z|c_1) \quad (12)$$

La densité de probabilité est entièrement déterminée par l'architecture du réseau qui permet de donner une expression à la densité conditionnelle $p(c_1|c_2)$ en utilisant les relations de voisinage sur la carte et la densité conditionnelle des observations $p(z|c_1)$. Chaque neurone représente en effet une loi normale qui permet d'exprimer la densité conditionnelle des observations $p(z|c_1) = \mathcal{E}_{c_1}(z, w_{c_1}, \sigma_{c_1})$. Si l'hypothèse que les relations de voisinage permettent de définir est considérée:

$$p(c_1|c_2) = [1/T_{c_2}] K^T(\delta(c_1, c_2)), \text{ avec } T_{c_2} = \sum_r K^T(\delta(c_2, r)) \quad (13)$$

Les densités de probabilité à posteriori des observations peuvent s'exprimer en fonction des distributions gaussiennes des différents neurones.

$$p_{c_2}(z) = [1/T_{c_2}] \sum_{r \in C_1} K^T(\delta(c_2, r)) \mathcal{E}_r(z, w_r, \sigma_r) \quad (14)$$

Ainsi, $p_{c_2}(z)$ apparaît comme un mélange local de densités gaussiennes qui fait intervenir tous les neurones de la carte. L'ensemble des vecteurs moyens $W = \{w_c, c \in C\}$ et les écarts-types $\sigma = \{\sigma_c, c \in C\}$ sont les paramètres qu'il faut estimer à l'aide de l'ensemble d'apprentissage A pendant la phase d'apprentissage. Grâce au formalisme probabiliste, il est possible de maximiser la vraisemblance classifiante de l'ensemble A .

On suppose que les observations de l'ensemble A sont indépendantes, que chaque observation z_i est engendrée par le générateur $p_{\chi(z_i)}$ qui est associé au neurone $X(z_i)$. On suppose aussi que les neurones c_2 de C_2 ont des probabilités a priori égales, la vraisemblance classifiante devient alors :

$$P(z_1, z_2, \dots, z_N | W, \sigma, X) = \prod_{i=1}^N p_{\chi(z_i)}(z_i) \quad (15)$$

Afin de maximiser cette expression par rapport aux paramètres du modèle W , σ et X , une manière classique consiste à minimiser l'opposé de la vraisemblance classifiante :

$$E(W, \sigma, X) = - \sum_{i=1}^N \text{Ln} \left[\sum_{r \in C} K_T(\delta(\chi(z_i), r)) f_r(z_i, w_r, \sigma_r) \right] \quad (16)$$

En utilisant le formalisme des nuées dynamiques, les deux phases d'affectation et de minimisation sont effectuées alternativement jusqu'à convergence :

- ❖ Phase d'affectation : On suppose que l'ensemble des paramètres W et celui des écarts-types σ sont constants et qu'ils prennent les valeurs courantes. Il faut minimiser E par rapport à la fonction d'affectation X . Il s'agit donc de trouver une nouvelle fonction d'affectation. Cette étape permet d'obtenir une nouvelle partition de l'ensemble des données D . La fonction d'affectation qui permet de minimiser E est celle qui consiste à affecter chaque observation z_i au neurone le plus probable selon la densité p_{c_2}

$$\chi(z) = \arg \max_{c_2} p_{c_2}(z) \quad (17)$$

- ❖ Phase de minimisation : Au cours de cette phase, on suppose que la fonction d'affectation est constante et égale à la fonction d'affectation courante. On cherche alors à minimiser $E(W, \sigma, X)$ par rapport à W et σ .

Les paramètres sont adaptés comme il en va pour la version globale de l'algorithme des cartes topologiques, en annulant les dérivés partielles de la fonctions $E(W^t, \sigma^t, X^t)$. Pour

résoudre l'équation, on utilise une procédure itérative qui suppose que pour la $i^{ième}$ itération, la valeur initiale des paramètres est assez proche des vraies valeurs. On obtient alors les formules de mise à jour suivantes :

$$w_r^t = \frac{\sum_{i=1}^N z_i K(\delta(r, \chi^{t-1}(z_i))) \frac{f_r(z_i, w_r^{t-1}, \sigma_r^{t-1})}{P_{\chi^{t-1}(z_i)}(z_i)}}{\sum_{i=1}^N K(\delta(r, \chi^{t-1}(z_i))) \frac{f_r(z_i, w_r^{t-1}, \sigma_r^{t-1})}{P_{\chi^{t-1}(z_i)}(z_i)}} \quad (18)$$

$$(\sigma_r^t)^2 = \frac{\sum_{i=1}^N \|w_r^{t-1} - z_i\|^2 K(\delta(r, \chi^{t-1}(z_i))) \frac{f_r(z_i, w_r^{t-1}, \sigma_r^{t-1})}{P_{\chi^{t-1}(z_i)}(z_i)}}{n \sum_{i=1}^N K(\delta(r, \chi^{t-1}(z_i))) \frac{f_r(z_i, w_r^{t-1}, \sigma_r^{t-1})}{P_{\chi^{t-1}(z_i)}(z_i)}} \quad (19)$$

Dans ces deux expressions, les paramètres à l'itération t s'expriment en fonction de ceux de l'itération $t-1$.

La complexité du modèle suppose que la minimisation est effectuée à partir de bonnes conditions initiales. Le modèle PRSOM peut être considéré comme une extension des modèles de cartes SOM. Il peut utiliser les paramètres estimés par ces modèles pour l'initialisation de l'ensemble des référents W .

L'algorithme PRSOM :

1. **Phase d'initialisation $t=0$:** les paramètres d'initialisation W^0 sont calculés en effectuant l'apprentissage avec un algorithme de cartes auto-organisatrices. La fonction d'affectation X^0 est définie à l'aide de l'équation (18) et σ^0 est calculé en appliquant (19). Le nombre maximal d'itérations N_{iter} est choisi.
2. **Etape itérative t :** W^{t-1} et σ^{t-1} étant connus et calculés à l'itération précédente :
 - ❖ Phase de minimisation : calcul des nouveaux paramètres W^t et σ^t selon l'équation (18) et (19)
 - ❖ Phase d'affectation : mise à jour de la fonction X^t associée à W^t et σ^t selon l'équation (17)
3. **Répéter** l'étape itérative jusqu'à atteindre $t > N_{iter}$ itérations ou une stabilisation de la fonction $E(W, \sigma, X)$

Le modèle PRSOM utilise un système de voisinage dont la taille décroît durant l'apprentissage à l'aide du paramètre T . A la fin de la phase d'apprentissage, la carte fourni

l'ordre topologique. La partition associée à la carte est définie à partir de la dernière fonction d'affectation $X^{N_{iter}}$. De même que pour les cartes SOM, l'ensemble D est divisé en M sous-ensembles, chaque neurone c de la carte représente un sous-ensemble $P_c = \{z / X^{N_{iter}}(z) = c\}$. Cette carte et cette partition ont été déterminées en tenant compte des distributions de probabilités. L'estimation des probabilités permet d'obtenir des informations supplémentaires qui peuvent être utilisées avec un profit dans des applications.

Ces informations sont en effet primordiales si ce modèle est utilisé pour traiter des problèmes de classification. L'estimation de la variance demande de prendre en considération toute la base d'exemples avant de modifier les différentes valeurs des paramètres.

L'algorithme PRSOM permet d'obtenir un grand nombre d'informations supplémentaires sur l'ensemble des observations étudié. Cependant, ce modèle ne peut être utilisé que si le nombre d'observations est assez grand pour permettre une estimation suffisamment précise des variances attachées aux gaussiennes

V Conclusion

L'apprentissage non supervisé permet d'affecter une observation à un sous-ensemble d'une partition, indépendamment de toute notion de classe. La partition proposée dépend de la densité de probabilité qui est sous-jacente aux observations de l'ensemble d'apprentissage. Elle est de nature géométrique et dépend du codage choisi pour représenter les observations.

Avec les cartes auto-organisatrices, le choix du codage introduit de manière implicite des informations sur le problème à traiter. Le principe de l'algorithme est de faire apparaître des groupements cohérents (les sous-ensembles de la partition) en relation avec l'application traitée. Pour chaque groupement, nous avons un représentant qui permet de résumer au mieux les données appartenant à sa partition.

L'algorithme des nuées dynamiques répète la phase itérative jusqu'à stabilisation de la fonction coût ou atteinte de la valeur N_{iter} . Dans le cas des réseaux de capteurs, le but est d'obtenir une valeur optimale de cette fonction de coût qui est la consommation de l'énergie, donc nous pouvons ne pas considérer N_{iter} sachant qu'il est initialisé à une valeur aléatoire, ce qui peut être déterminant dans la convergence de l'algorithme.

Cependant, nous avons vu comment que le paramètre T (voisinage) pouvait affecter directement la qualité du clustering résultant, c'est pour cette raison que l'algorithme d'optimisation globale des cartes topologiques s'avèrent relativement plus adéquats du fait qu'il exécute la minimisation un certain nombre de fois en faisant décroître la valeur de T dans l'intervalle $[T_{\min}, T_{\max}]$. Cependant, cet algorithme itère jusqu'à atteindre $T = T_{\min}$, il faut alors affecter cette variable de manière à ce qu'elle soit la plus adéquate possible pour atteindre un clustering efficace.

L'algorithme de Kohonen est similaire à l'algorithme d'optimisation globale des cartes topologiques, à la différence que les fonctions d'affectation sont distinctes et à chaque itération, une seule observation est choisie de manière aléatoire pour mettre à jour les valeurs des référents. Cet algorithme constitue une version stochastique de celui des cartes topologiques. Il utilise une seule observation par itération (choisie aléatoirement) pour mettre à jour les référents. De plus, la fonction d'affectation est fixe (ne sera pas réaffectée) et la formule de mise à jour des référents est à base du gradient qui décroît au cours des itérations.

Les cartes auto-organisatrices font de même sous forme d'un réseau de neurones constitués de deux couches. La couche d'entrée (observations) et la couche de compétition (neurones). Les vecteurs référents concernant un neurone c forment le vecteur des connexions (poids) qui arrivent à c . Ces poids sont adaptatifs tels qu'ils sont mis à jour avec l'une des règles présentées précédemment. De plus, l'adaptation des paramètres n'est faite que sur les régions actives qu'une observation z peut affecter ($\min(\|z - w_c\|^2)$). C'est cette notion de voisinage qui introduit les contraintes topologiques dans la représentation finale. Le problème majeur de ces cartes revient au choix de la structure de carte (à priori) bien adaptée au problème traité. Un tel choix est fait indépendamment de la structure interne des données.

Souvent, les observations n'occupent pas tout l'espace des données tel que pour les différentes régions, les concentrations des données sont variables. Afin de synthétiser cette répartition, nous pouvons utiliser des fonctions densités de probabilité d'où les algorithmes d'optimisation probabiliste à savoir PRSOM qui attribut à chaque neurone c de la carte, une fonction densité normale. Mais ce genre de modèle reste assez complexes à utiliser et leur application se rapporte aux domaines dont la volumétrie des données est très importante.

Notre objectif est de pouvoir exploiter les cartes auto-organisatrices dans le clustering des réseaux de capteurs, qui reste jusqu'à nos jours, une phase coûteuse en terme de

consommation d'énergie. L'utilisation de ces cartes permettra d'avoir un clustering optimal qui minimise au mieux la consommation d'énergie.

Nous pouvons faire un rapprochement entre les cartes SOM et les réseaux de capteurs dans le sens où :

- Les observations (x_i) sont les nœuds du réseau.
- Les référents (w_j) qui représentent et qui résument un ensemble d'observations sont les têtes de clusters.
- La couche de compétition contient les nœuds qui se proposent aux élections pour devenir têtes de groupes.

Quelques travaux ont été menés dans ce sens, à savoir LEA2C, CODA...ect. Le paragraphe suivant est dédié à la présentation de ces deux protocoles qui sont à base des cartes SOM.

VI Quelques protocoles de routage à base de cartes SOM

VI.I Le protocole LEA2C (Low Energy Adaptive Connectionist Clustering)

LEA2C est une extension du protocole LEACH et dans lequel les auteurs ont proposé d'utiliser quelques particularités du protocole LEACH-C.

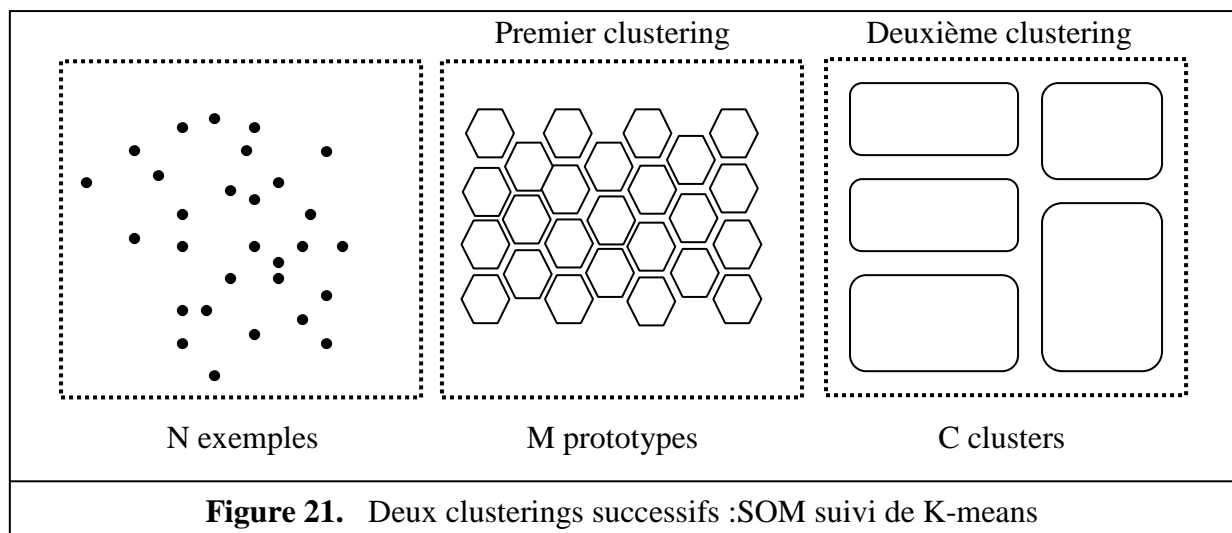
Dans LEA2C, les nœuds disposent d'un système de localisation géographique « GPS » (Global Positioning System). Ce protocole peut alors être vu comme un protocole de routage géographique.

L'algorithme est itératif tel qu'à chaque itération, le nœud puit calcule le clustering en fonction des coordonnées des nœuds vivants et pour chaque cluster, elle crée une table de fréquence TDMA et lui affecte un code aléatoire d'une liste de codes de propagation CDMA. Les techniques TDMA et CDMA permettent d'éviter les interférences. Les nœuds peuvent éteindre leurs antennes hors l'intervalle TDMA qui leur sera affecté [Dehni et al.05].

Le nœud puit attribut ensuite à chaque nœud (tête de cluster ou un simple capteur) un rôle en lui envoyant son code et sa fréquence de communication.

Le clustering des nœuds capteurs se fait en deux étapes qui sont :

- Clustering avec les cartes SOM : qui est une phase préliminaire dans le but d'effectuer une sorte de prétraitement des données (réduction de la dimension, regroupement, visualisation,...). Ceci permet de réduire la complexité de calcul des clusters et d'assurer une meilleure visualisation des résultats de la classification automatique. Ainsi, les cartes SOM permettent de réduire la dimension de l'espace des données et de visualiser les clusters dans le plan.
- Le résultat de la première étape est optimisé en utilisant l'algorithme de k-means qui est une autre méthode de classification. Son principe consiste à choisir arbitrairement une partition, ensuite, les exemples sont examinés un par un. Si un exemple devient plus proche du centre d'une classe autre que la sienne, il est déplacé vers cette nouvelle classe. Dans ce cas, les centres des nouvelles classes sont recalculés et les exemples sont réaffectés aux partitions et ainsi de suite jusqu'à avoir des partitions stables.



L'algorithme de base de la méthode des k-means nécessite à priori de fixer une valeur pour k qui représente le nombre de clusters souhaités. Cependant, il existe un algorithme qui permet de calculer une meilleure valeur pour k et pour laquelle le clustering est optimal. Son principe de base est la minimisation de l'indice de Davies-Bouldin [David et al. 79] défini comme suit :

$$I_{DB} = \frac{1}{C} \sum_{k=1}^C \max_{l \neq k} \left\{ \frac{S_c(Q_k) + S_c(Q_l)}{d_{cl}(Q_k, Q_l)} \right\} \quad (20)$$

Avec

$$S_c(Q_k) = \frac{\sum_i \|x_i - c_k\|^2}{|Q_k|} \quad (21)$$

$$d_{cl}(Q_k, Q_l) = \|c_k - c_l\|^2 \quad (22)$$

Où C est le nombre de cluster, S_c est la dispersion intra-cluster et d_{cl} est la distance entre les centres des clusters k et l .

Le choix des têtes de clusters peut être fait selon l'un des trois critères suivants :

- ❑ Le nœud ayant la quantité d'énergie maximale.
- ❑ Le nœud étant le plus proche du centre de gravité du cluster.
- ❑ Le nœud étant le plus proche du puit.

Ces itérations sont répétées jusqu'à ce qu'il ne reste aucun nœud dans le réseau.

Les résultats de simulation ont montré que LEA2C et avec le meilleur critère de choix des têtes de cluster permet d'assurer une survie de la totalité des nœuds durant 90% de la durée de vie de tout le réseau. Comparé à LEACH et à LEACH-C, LEA2C offre un gain de 30 fois plus important en terme de durée de vie globale du réseau et 40 fois plus dans le cas où la tête de cluster retenue est le nœud le plus proche de la station de base parmi les nœuds de chaque cluster [Dehni et al.05].

Bien que LEA2C ait considérablement amélioré la durée de vie du réseau global et la survivabilité des nœuds du réseau pendant 90% de la vie du réseau, cependant, il possède certaines limites qui sont :

- L'utilisation du système GPS est très coûteuse.
- La centralisation du traitement du clustering qui est un processus itératif et dont la fréquence est assez élevée, ceci engendre un grand nombre de messages échangés entre la station de base et les nœuds du réseau. Lorsque le nombre de nœuds devient important, le processus de clustering engendre une grande surcharge du réseau.
- De plus, la station de base peut attribuer des rôles aux nœuds (tête de cluster ou simple) par rapport à la position géographique sans aucune connaissance sur les relations de voisinage entre les nœuds. Ainsi, un nœud simple peut appartenir à un cluster sans pouvoir communiquer avec sa tête de cluster (par exemple présence d'obstacle) entre ces deux derniers.

VI.II Le protocole CODA (Cluster-based self-Organizing Data Aggregation) [SangHak et al.05]

Dans [SangHak et al.05], les auteurs ont proposé d'utiliser les cartes SOM pour réaliser un clustering des nœuds par rapport aux données elles mêmes après avoir appliqué un premier regroupement à base de la distance euclidienne (fait à l'initialisation du réseau). Ceci permet à la fois de réduire la consommation d'énergie et surtout d'améliorer la qualité de l'information produite par le réseau de capteurs. Son but principal est alors de trouver un algorithme de clustering pour transmettre les données de manière efficace et sans pour autant dégrader la qualité de l'information produite.

Les implémentations des cartes SOM dans leur forme originale sont principalement basées sur un seul processus. Cependant, les réseaux de capteurs sont caractérisés par la densité de leurs nœuds et par conséquent, ça serait difficile d'implémenter les cartes SOM dans une approche centralisée. Cependant, les concepteurs de CODA ont proposé d'apporter des modifications aux cartes SOM traditionnelles afin de s'adapter au mieux aux caractéristiques des réseaux de capteurs.

Dans les cartes SOM classiques, les entrées de toutes les unités (neurones) sont exactement les mêmes. Dans CODA, l'idée est de classer les données captées et de les regrouper en utilisant les cartes auto-organisatrices de Kohonen. Ils choisissent les têtes de clusters qui gèrent le scheduling des communications, fusionnent et partitionnent les clusters. Le réseau de capteurs est déployé directement en clusters géographiques, ensuite, le réseau est partitionné à base des données récoltées.

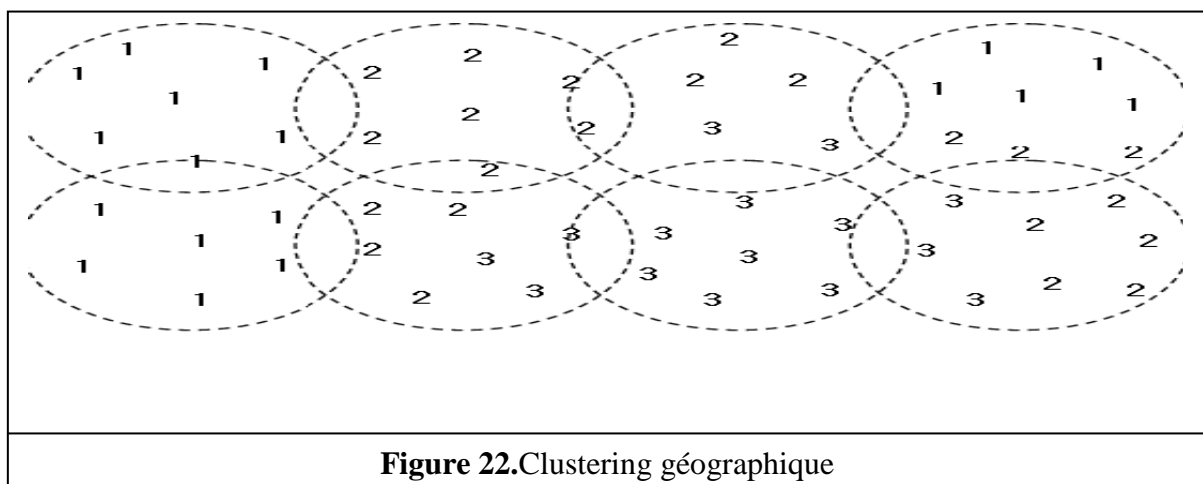
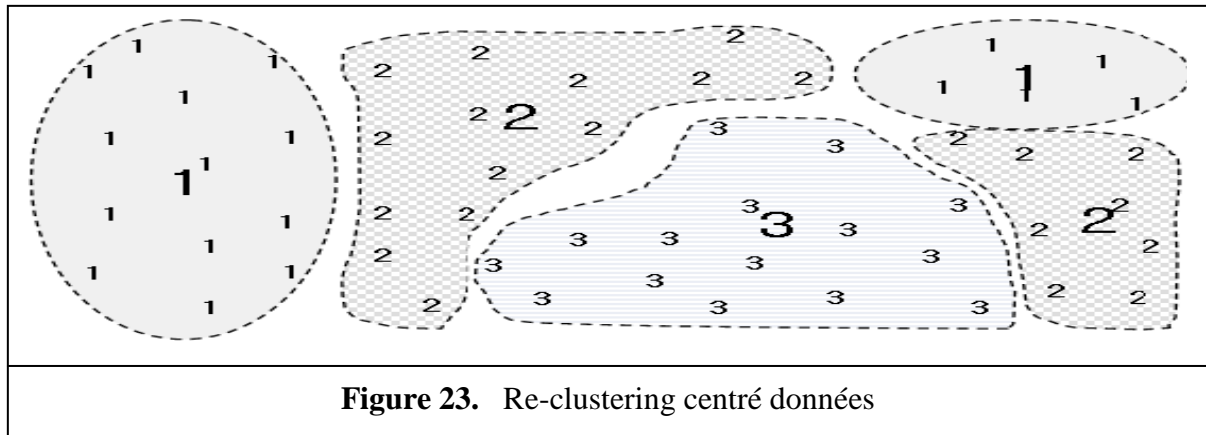


Figure 22.Clustering géographique



Dans CODA, les auteurs ont proposé de déployer les nœuds avec assez d'apprentissage sur les poids w'_j , contrairement aux cartes SOM qui le font de manière aléatoire. Un nœud sensor peut être vu comme étant un agent intelligent tel que sa durée de mise en veille va décider la qualité des données, la durée de vie du réseau, l'adaptabilité du nœud capteur et le niveau de collaboration on-line.

Deux méthodes de clustering sont proposées dans CODA. Une tête de cluster initiale schedule la période de fusion de données locales. Chaque nœud simple transmet les données à sa tête de cluster dans le quanta de temps qui lui est approprié. La tête de cluster identifie le vainqueur, met à jour le vecteur prototype (le référent) et calcule les erreurs. Un algorithme de partitionnement est exécuté après que le temps d'agrégation de données ait expiré. L'algorithme de fusion entre les clusters voisins qui ont des données similaires sera par la suite activé.

- Algorithme de partitionnement : Cet algorithme est exécuté lorsque l'erreur dans un cluster devient importante, ce qui s'explique par deux cas possibles qui sont :
 - ✓ Soit il y a eu un changement brusque dans l'environnement.
 - ✓ Ou alors il y a différents types de données dans quelques unités

L'erreur est calculée à base des valeurs de données récoltées, elle représente la différence entre ces valeurs et elle est calculée à l'aide des formules suivantes :

$$\text{L'erreur dans un groupe } G_i : \varepsilon_i = \frac{|m_i - m'_i|}{m_i} * 100 \quad (23)$$

tel que m et m' sont les seuils des valeurs de données d'un groupe.

$$\text{L'erreur globale est : } \delta = \frac{1}{n} \sum_{i=1}^n \delta_i \quad (24)$$

L'algorithme de partitionnement est activé dans le second cas. Dès que la tête de cluster apprend quel est le (ou les) nœud (s) qui est (sont) à l'origine de l'erreur, il peut partitionner son cluster à base de cette valeur d'erreur.

- Algorithme de fusion : Après le partitionnement interne du cluster, un algorithme de fusion entre les clusters voisins est exécuté. Chaque tête de cluster diffuse la valeur de ses données agrégées. Deux clusters sont fusionnés si la différence entre leurs données agrégées est inférieure à un certain seuil.

La qualité du réseau est basée sur la qualité de ses données agrégées. La valeur du seuil dépend du type d'application et constitue l'un des facteurs de conception des réseaux de capteurs.

Les simulations dans CODA ont prouvé une amélioration considérable de la qualité de l'information retournée par le réseau de capteurs. Cependant, l'utilisation de ce protocole s'avère limitée dans le cas où l'environnement étudié ne connaît pas beaucoup de variété de valeurs de données, ce qui mène à l'obtention de clusters qui soient assez grand et dont la gestion devient cruciale et coûteuse en énergie.

De plus, le traitement itératif pour la construction du cluster risque d'épuiser les têtes de clusters rapidement si les erreurs se produisent de manière fréquente.

VII Synthèse

Le tableau suivant est un récapitulatif de tous les protocoles présentés dans les sections précédentes par rapports à différents paramètres importants dans les réseaux de capteurs :

	TECHNIQUE DE ROUTAGE DE DONNEES	LOCALISATION GEOGRAPHIQUE	ORIENTATION DONNEES	QUALITE DE SERVICE	MODE DE REPORTAGE DE DONNEES	CONSOMMATION D'ENERGIE	SCALABILITE	TOLERANCE AUX PANNES	MOBILITE	AGREGATION DE DONNEES
GAF	A base de sauts / Hiérarchique	Oui	Non	Limité	A base de requête	Economique	Bon	Limité	supportée	Non
LEACH	Hiérarchique	Non	Non	Limité	Continu	Moyen	Bon	Limité	Nœuds fixes	Oui
PEGASIS	Hiérarchique /	Non	Non	Limité	Continu	Economique	Bon	Bon	Nœuds fixes	Oui

	A base de sauts									
TEEN	Hiérarchique	Non	Non	Limité	A base d'événement	Economique	Bon	Limité	Nœuds fixes	Oui
APTEEN	Hiérarchique	Non	Non	Limité	A base d'événement / Continu	Economique	Bon	Limité	Nœuds fixes	Oui
LEA2C	Hiérarchique	Oui	Non	Limité	Continu	économique	bon	Limité	Nœuds fixes	Oui
CODA	Hiérarchique	Oui	Oui	moeyenne	Continu	Moyen	bon	bon	Nœuds fixes	Oui

Tableau 2. Synthèse des protocoles de routage

Dans notre solution, nous proposons de comparer notre travail au protocole LEA2C, vue qu'il utilise les cartes SOM dans la tâche de clustering. Ce choix est fait aussi par rapport aux résultats de simulations obtenus dans cette solution par rapport à LEACH.

VIII Conclusion

Etant basées sur des algorithmes à apprentissage non supervisé et sur des théories mathématiques et probabilistes robustes, les cartes auto-organisatrices de Kohonen constituent un moyen de classification très puissant.

En effet, le clustering dans les réseaux de capteurs s'avère l'approche la plus intéressante pour optimiser la consommation d'énergie des capteurs et pour améliorer la scalabilité. Cependant, elle reste une phase assez complexes à réaliser du fait que les nœuds n'ont aucune connaissance sur la structure globale du réseaux de capteurs.

Il serait très intéressant d'exploiter les cartes SOM afin d'optimiser cette phase de construction de clusters. Pour ce faire, il faut ajuster ces cartes afin de s'adapter au mieux aux caractéristiques et contraintes des réseaux de capteurs, tel que leur déploiement dans leur version originale de manière centralisée ou distribuée, risque de poser certains problèmes, à savoir :

- De manière centralisée : leur traitement devient complexe à cause de la densité du réseaux de capteurs. De plus leur exécution peut fournir des affectations logiques entre les nœuds qui ne soient pas cohérentes avec les liens physiques entre les capteurs.
- De manière distribuée : une exécution itérative de l'algorithme de Kohonen dans les réseaux de capteurs risque d'épuiser leurs nœuds rapidement.

Dans ce présent travail, nous proposons d'utiliser les cartes SOM dans le processus de clustering afin d'optimiser la consommation d'énergie et d'améliorer la qualité de l'information produite par le réseau de capteurs. Le chapitre suivant sera dédié à la présentation de notre nouveau protocole SOBC (Self Organizing Based Clustering).

Troisième chapitre

Troisième chapitre : **SOBC (Self Organizing Based Clustering)**

SOBC (Self Organizing Based Clustering)

I Introduction

Comme nous l'avons déjà mentionné auparavant dans ce document, le déploiement des réseaux de capteurs est contraint par plusieurs limitations notamment l'insuffisance énergétique et la densité de ses nœuds. Ce qui rend les protocoles de routage classiques non satisfaisants lorsque le réseau passe à un large déploiement.

Le clustering dans les réseaux sensor s'avère incontournable afin de mener à bien le processus de routage de données et d'assurer la scalabilité. Dans la littérature, nous pouvons distinguer plusieurs méthodes de clustering dont la plus connue qui consiste à établir une structure arborescente reliant les nœuds du réseau. Le problème majeur de cette méthode est le processus de maintenance de cette structure, tels que dans les protocoles de routage arborescents, les auteurs préfèrent reconstruire cette structure lorsqu'il y a une cassure sur l'un de ses liens au lieu de la maintenir, ce qui est très coûteux en terme d'énergie.

Les cartes SOM sont des modèles de classification non supervisés, efficaces et très utilisées dans divers domaines. Elles sont basées sur des algorithmes de clustering qui convergent vers un minimum optimal d'un paramètre donné.

Dans notre travail, nous proposons d'exploiter les cartes SOM dans le processus de clustering afin d'améliorer au mieux le rendement des réseaux de capteurs, tout en garantissant la qualité de l'information retournée et en optimisant la consommation d'énergie et par conséquent, augmenter la durée de vie globale du réseau.

II Problématique

Tenant compte des contraintes liées aux capacités des nœuds capteurs, l'utilisation des techniques de l'intelligence artificielle a été écartée pendant les premières années de recherche dans ce domaine. Cependant, l'idée d'utiliser les algorithmes d'auto-organisation dans la procédure de clustering des réseaux sensor a été proposée dans certains travaux récents. Ceci est dû au fait que les principes et les buts du regroupement des réseaux de capteurs et de ces algorithmes soient convergents.

L'exploitation des algorithmes d'auto-organisation a été faite selon deux approches différentes, soit de manière centralisée (par exemple LEA-2C) ou alors distribuée (par exemple CODA).

Dans la première approche, il s'agit d'envoyer les informations nécessaires au clustering (position géographique) par les nœuds sensors à la station de base, qui à son tour, exécute l'algorithme de Kohonen et établit la structure du regroupement. Enfin, la station de base envoie aux nœuds capteurs leurs rôles (tête de cluster ou un simple membre). Dans cette approche, un nombre important de messages sont échangés dans le réseau et de manière fréquente rien que pour s'organiser en groupes, ceci est coûteux sur le plan énergétique, ainsi que sur le temps de réponse du processus de clustering.

De plus, il est important d'avoir une idée sur les relations de voisinage entre les nœuds, car parfois et bien que des nœuds soient proches physiquement, le lien de communication entre eux n'est pas forcément fiable.

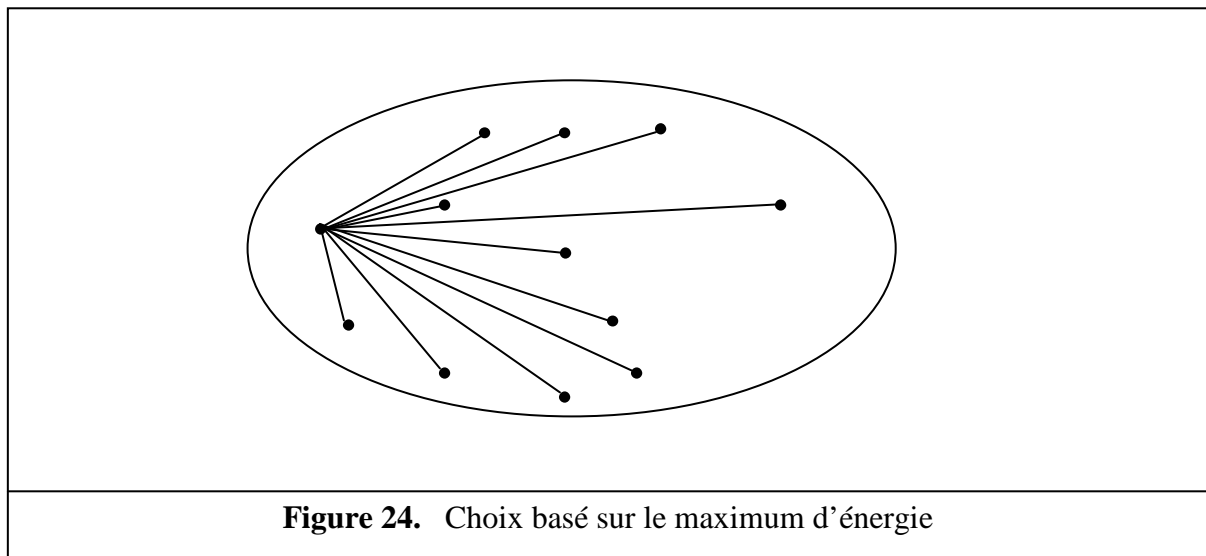
L'approche distribuée permet de fragmenter et de paralléliser le traitement, améliorant ainsi, le temps de réponse de la procédure de clustering et fournissant une structure de liens physique fiable (concordante avec les liens réels entre les capteurs), mais qui risque aussi d'épuiser les nœuds plus rapidement.

Il serait alors judicieux d'équilibrer l'utilisation des cartes SOM afin d'améliorer la qualité de l'information retournée sans pour autant, trop dégrader les capacités énergétiques.

Le processus de clustering est une étape très sensible et coûteuse dans la mise en production du réseau de capteurs. Cependant, dans la plupart des protocoles de routage proposés pour ces

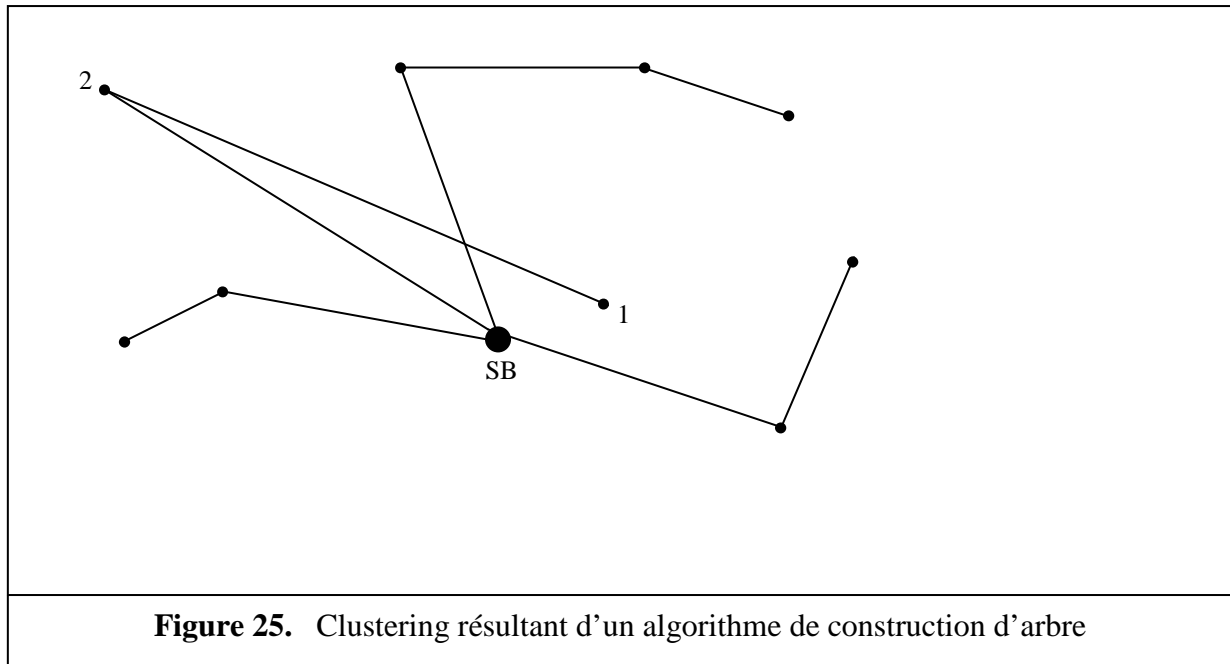
réseaux, reposent sur la notion de round et d'autres, exécutent continuellement cette procédure. En effet il faut réduire cette fréquence d'exécution autant que possible, car par exemple, si un réseau est plus ou moins stable, ça ne sert à rien de réaffecter les rôles si les têtes de groupes actuelles possèdent encore plus de capacités que les autres nœuds afin d'exécuter leurs tâches. Ou alors, si un imprévu a surgit dans un groupe donné et qu'il y a nécessité de réaffecter les rôles dans ce groupe, il n'est pas nécessaire de relancer ce processus pour tout le réseau, mais juste pour le groupe concerné ou à la limite pour les clusters qui sont dans son voisinage.

Un autre problème est lié au critère de choix de la tête de cluster. Souvent, ce choix est fait par rapport à l'énergie maximale disponible, à la position géographique ...etc. Ceci risque d'épuiser rapidement les nœuds répondant à ces critères. Par exemple, si nous considérons un choix lié au nœud ayant le maximum d'énergie, nous pouvons avoir des liens de clustering comme montré dans la figure suivante :



Dans ce cas de figure, nous risquons d'épuiser rapidement les nœuds du groupe, car la tête de cluster est assez loin des autres nœuds, ce qui oblige les autres nœuds à émettre avec une puissance plus importante afin d'atteindre la tête de cluster.

Comme nous pouvons avoir des liens de routage qui ne soient pas économes en énergie, la figure suivante montre un clustering résultant d'un algorithme de construction d'un arbre :



Les messages du noeud (1) sont routés par le noeud (2) afin d'atteindre la station de base malgré que le capteur soit dans le périmètre de la portée de la station de base.

Il serait intéressant de trouver des moyens de sélection des têtes de groupes de sorte à équilibrer les charges sur l'ensemble des nœuds et à gagner sur les plans de survivabilité maximale et de qualité de l'information fournie par le réseau. Il s'agit alors de trouver une solution juste moyenne, qui synthétise un compromis entre les objectifs attendus des réseaux de capteurs et de leurs contraintes.

Dans ce qui suit, nous présenterons notre solution dans laquelle, nous considérons le même modèle de communication utilisé dans les protocoles LEACH et LEA2C. Un capteur utilise son énergie pour réaliser trois actions principales : l'acquisition, la communication et le traitement des données [Dehni et al.05].

- L'énergie consommée pour effectuer l'acquisition n'est pas très importante. Néanmoins, elle varie en fonction du phénomène et du type de surveillance effectuée.
- Les communications consomment beaucoup plus d'énergie que les autres tâches. Elles couvrent les communications en émission et en réception.
 - Pour transmettre un message de k bits sur une distance de d mètres, l'émetteur consomme :

$$E_{TX}(k, d) = \begin{cases} k.E_{elec}(k, d) + k.\varepsilon_{friss}.d^2 & Sid < d_{crossover} \\ k.E_{elec}(k, d) + k.\varepsilon_{rwo_ray_amp}.d^4 & Sinon \end{cases}$$

- Pour recevoir un message de k bits, le récepteur consomme :

$$E_{RX}(k) = E_{RX_elec}(k) = k * E_{elec}$$

E_{elec} : représente l'énergie de transmission/réception électronique et E_{TX_amp} : indique l'énergie d'amplification. ε_{amp} : est le facteur d'amplification et $d_{crossover}$: représente la distance limite pour laquelle les facteurs de transmission changent de valeur.

- L'énergie consommée pour les opérations de calculs est beaucoup plus faible que l'énergie de communication. L'énergie nécessaire pour transmettre 1 KB sur une distance de 100m est approximativement équivalente à l'énergie nécessaire pour exécuter 3 millions d'instructions avec une vitesse de 100 millions d'instructions par seconde (MIPS).

III SOBC (Self Organizing Based Clustering)

Un clustering intelligent, une survivabilité maximale, une information de qualité...etc sont des objectifs majeurs à atteindre lorsqu'un réseau de capteurs est déployé. Bien que certains critères s'avèrent divergents, par exemple assurer la qualité de service et économiser l'énergie, il faut cependant, réaliser un équilibre dans le choix des solutions.

Dans notre protocole, nous proposons d'utiliser les cartes SOM afin d'implémenter la structuration du réseau en clusters avec le déploiement d'un minimum d'intelligence. L'intérêt principal de ce choix revient à la construction dynamique de la topologie du réseau tout en optimisant le choix des têtes de clusters et en répartissant les traitements.

La première question qui se pose est quelle version des algorithmes d'auto-organisation, parmi celles présentées précédemment dans l'état de l'art, peut-t-on utiliser dans notre solution tout en tenant compte de la nature des réseaux de capteurs.

Les algorithmes des nuées dynamiques, d'optimisation globale et de Kohonen ne sont pas adaptatifs chacun (l'espace de compétition est fixe), ce qui n'est pas le cas dans les réseaux de capteurs, tel que l'espace de compétition doit être variable et dépendra des nœuds vivants dans le système.

Quant à l'algorithme d'optimisation probabiliste, il reste assez complexe à appliquer aux réseaux de capteurs car il engendre beaucoup de traitements et de calculs. De plus, leur utilisation n'a de sens que sur un nombre très important d'observations, ce qui n'est pas intéressant dans notre cas, puisque nous proposons de répartir le traitement de clustering. Ainsi, nous aurons à calculer le clustering pour des sous ensembles du réseau.

Cependant, les algorithmes d'optimisation adaptative des cartes topologiques s'avèrent plus adéquats pour le clustering des réseaux de capteurs. Le problème majeur des ces algorithmes adaptatifs, réside dans la définition d'une structure adéquate et pertinente de la couche de compétition. Dans notre solution, nous proposons de définir à chaque exécution, une structure de carte identique à l'image actuelle du réseau (nœuds vivants réellement dans le réseau). Ceci s'explique par le fait que seuls ces capteurs vivants peuvent concurrencer le rôle de tête de groupe et ils doivent tous contribuer à cette tâche de clustering.

Les algorithmes d'optimisation adaptative des cartes topologiques utilisent l'un des algorithmes d'auto-organisation définis dans l'état de l'art, c'est la structuration des espaces de compétition qui est différente (réseau à deux couches, de compétition et d'observation).

En effet, l'algorithme de Kohonen utilise une seule observation (choisie aléatoirement) par itération, ce qui risque de faire converger l'algorithme de différentes manières selon l'effet des nœuds choisis durant les différentes itérations. Dans notre cas, tous les nœuds doivent affecter leurs voisinages afin de construire les clusters. Par conséquent, l'algorithme d'optimisation globale s'avère la solution la plus adéquate du moment où il fait décroître le paramètre de voisinage T jusqu'à l'obtention d'une fonction de coût stable (grâce à la phase d'initialisation qui exécute l'algorithme des nuées dynamiques) et d'un clustering de bonne qualité. Cependant, il faut faire attention aux différents paramètres de ces cartes et à leurs initialisations.

III.I Personnalisation et paramétrage des cartes SOM

Telles que les cartes SOM sont définies, elles ne peuvent pas être directement appliquées au clustering des réseaux de capteurs vu les caractéristiques contraignantes de ces derniers. Il faut alors personnaliser certains de leurs paramètres.

III.I.1 Liens entre les couches d'observations et de compétition

Dans la version originale des cartes SOM, tous les nœuds de la carte (couche de compétition) sont entièrement liés à la couche d'entrée. Dans notre protocole, nous proposons de réduire ce nombre de liens afin de ne représenter que ceux qui reflètent les relations de voisinage entre les nœuds. Cette représentation nous permet à la fois :

- D'éviter l'affectation de nœuds simples à des têtes de clusters qui ne leur soient pas accessibles car dans les cartes SOM traditionnelles, un nœud de la carte est représentant d'un ensemble d'observations de la couche d'entrée. Par conséquent, si deux observations sont proches en valeurs et issues de nœuds très distants, leurs nœuds émetteurs seront affectés à un même point de la carte, ou autrement dit, à la même tête de cluster.
- De casser les grands clusters pour répartir les charges et pour éviter d'épuiser rapidement leurs têtes.

- De réduire le traitement des cartes SOM autant que possible, car un nombre important de liens seront éliminés, ainsi que les traitements de mise à jour et de calcul des référents (vecteurs de poids) qui en dépendent.

Avec cette manière d'exploitation des cartes auto-organisatrices, le clustering résultant peut être obtenu aussi bien de manière centralisée que distribuée. Ceci est dû à l'inclusion des relations de voisinage entre les nœuds.

III.I.2 Dimension de la couche de compétition

La définition de la dimension de la carte (couche de compétition) constitue une tâche complexe à réaliser. Souvent, les concepteurs sont menés à faire appel à des connaisseurs du domaine étudié afin de réaliser cette tâche.

Dans notre cas de partitionnement du réseau de capteurs et tenant compte de sa nature, nous proposons d'utiliser une carte contenant une copie identique des capteurs réels existants sur le terrain. Autrement dit, la dimension de la carte de compétition est égale à celle de la couche d'entrée (correspond au nombre de nœuds vivants dans un sous ensemble du réseau).

En effet, ce choix s'explique par le fait que tout nœud du réseau peut être tête de cluster si son état est meilleur par rapport à celui de son voisinage. Par conséquent, l'ensemble des nœuds doit figurer dans la couche de compétition.

III.I.3 Paramètre à optimiser

Un autre point très important dans la conception des cartes SOM est le paramètre à minimiser. Généralement, afin de définir la tête de cluster, les protocoles de routage se basent sur la position géographique ou alors sur les nœuds ayant le maximum d'énergie, ceci engendre l'affaiblissement rapide des nœuds qui vérifient ces conditions. Dans notre protocole, nous proposons d'effectuer un équilibre dans ce choix. Il s'agit de minimiser le critère suivant :

$$c = \delta_{val} \frac{nv}{e} ,$$

Tel que n_v et e représentent respectivement, le nombre de voisins et la quantité d'énergie restante. Quant à δval , elle synthétise le décalage ou la différence entre la valeur moyenne captée par un nœud donné et celles de son voisinage.

Le rapport $\frac{n_v}{e}$ permet d'exprimer la relativité entre le nombre de voisins d'un nœud donné et sa capacité en énergie. Il augmente avec l'évolution du nombre de voisins par rapport à l'énergie restante. Sa minimisation pour le choix de la tête de cluster permettra d'assurer l'équilibre et la répartition des charges sur les têtes de groupes du réseau tel que pour qu'un nœud vainqueur devienne tête de cluster, il doit posséder une valeur moyenne en énergie par rapport à son voisinage.

δval permet de trancher à quelle tête de cluster sera associé un nœud précis dans le cas où nous aurons deux affectations trop proches en valeurs. En effet, ceci a un autre apport par rapport à la qualité de l'information retournée. Autrement dit, plus un cluster regroupe des nœuds fournissant des données cohérentes et concordantes, plus la qualité de l'agrégation est améliorée.

III.I.4 Initialisation des référents et du nombre d'itérations de l'algorithme d'auto-organisation

Dans l'algorithme d'optimisation globale, les référents et N_{iter} sont initialisés de manière aléatoire, ces référents sont mis à jour au fur et à mesure que l'on boucle dans cet algorithme, alors que la valeur de N_{iter} est fixe. Dans notre protocole, nous procédons de la même manière juste pour la première exécution de l'algorithme de clustering. Par ailleurs, nous proposons :

- D'utiliser un minimum d'intelligence tel que les vecteurs de poids seront initialisés avec les valeurs finales de l'exécution précédente de l'algorithme de Kohonen. Ceci nous permettra de réduire le nombre d'itérations du processus de clustering en anticipant la stabilisation de la fonction de coût J_{som}^T et ainsi, terminer ce processus.
- Pendant la première exécution de l'algorithme de Kohonen, N_{iter} est initialisé de manière aléatoire. A la fin de cette exécution et une fois la fonction de coût est stable, nous pouvons déduire le nombre d'itérations approximatif et nécessaire pour cette stabilité. Ce

nombre sera utilisé comme valeur d'initialisation des prochaines exécutions des cartes SOM.

En effet, les cartes SOM bouclent jusqu'à l'obtention de valeurs stables de la fonction coût, ceci est réalisé généralement après un nombre assez grand d'itérations (de l'ordre de 1000). Dans le cas de réseaux de capteurs qui ne sont pas trop mobiles, la structure du réseau a tendance à rester inchangeable pendant une bonne partie de sa durée de vie (jusqu'à épuisement des nœuds).

De ce fait, l'initialisation des référents avec les dernières valeurs de l'exécution précédente, nous permet d'obtenir une stabilité beaucoup plus rapide de la fonction coût surtout si la structure du réseau n'a pas trop changé. Ainsi, le nombre d'itérations va diminuer de manière considérable et le nœud responsable de l'exécution de l'algorithme d'auto-organisation va consommer beaucoup moins d'énergie comparant à l'utilisation de la version classique de cet algorithme.

Pour ce faire, il est nécessaire d'introduire la notion du contexte qui constitue l'ensemble des connexions des différents nœuds à leurs chefs de groupes. Il peut être utilisé pour la phase d'initialisation de l'algorithme de clustering dans sa prochaine exécution.

Lorsqu'un même nœud calcule le clustering d'un même groupe plusieurs fois, nous n'avons pas besoin d'échanger ce contexte entre les capteurs. Dans le cas contraire, cet échange s'avère nécessaire. Pour ce faire, nous pouvons distinguer deux approches différentes :

- Soit la tête de cluster actuelle envoie tous les vecteurs de poids au nouveau capteur qui sera responsable de calculer le clustering prochain.
- Soit les nœuds du cluster envoient chacun son vecteur de connexion (aux différents capteurs) au nouveau nœud qui va calculer le clustering. Ce dernier pourra construire le contexte à travers ces vecteurs de poids. C'est cette nouvelle option qui nous intéresse dans la suite de ce document.

III.I.5 Définition du paramètre T et de sa loi de décroissance

Comme nous l'avons déjà défini dans le deuxième chapitre, le paramètre T reflète la notion de voisinage dans laquelle, un nœud x peut avoir une influence. Ce paramètre varie dans un intervalle $[T^{\min}, T^{\max}]$ telle que sa décroissance nous permet d'avoir une répartition meilleure et uniforme des neurones vainqueurs par rapport à la répartition des nœuds (observations).

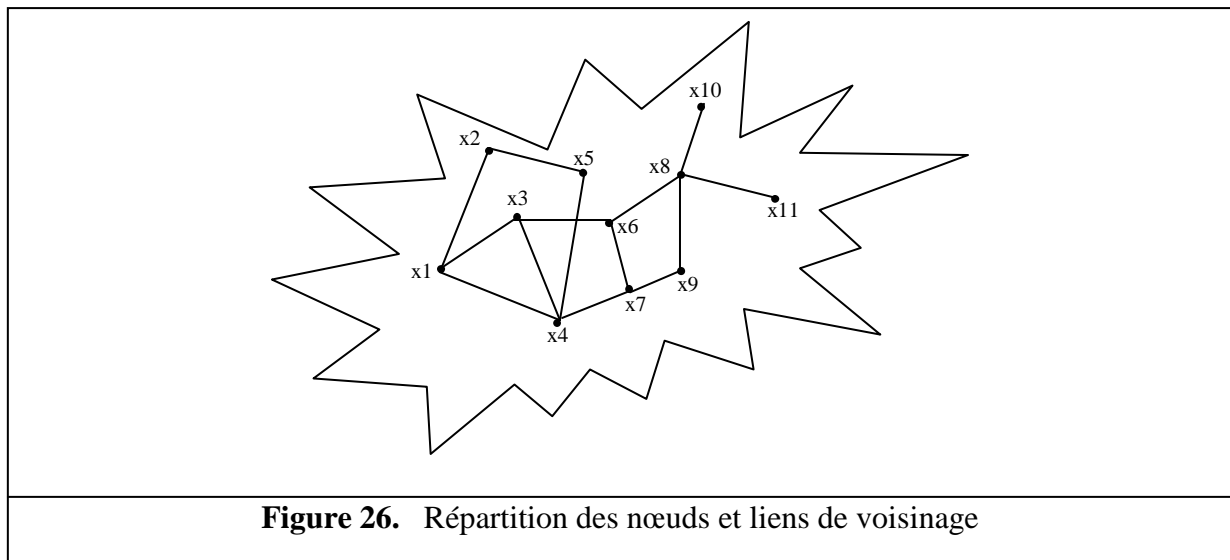
En effet, l'évolution de T est proportionnelle à celle des valeurs des données captées dans le sens où, plus le voisinage (ou la distance) est grand, plus la probabilité que les données captées (δval) soient différentes devient plus importante.

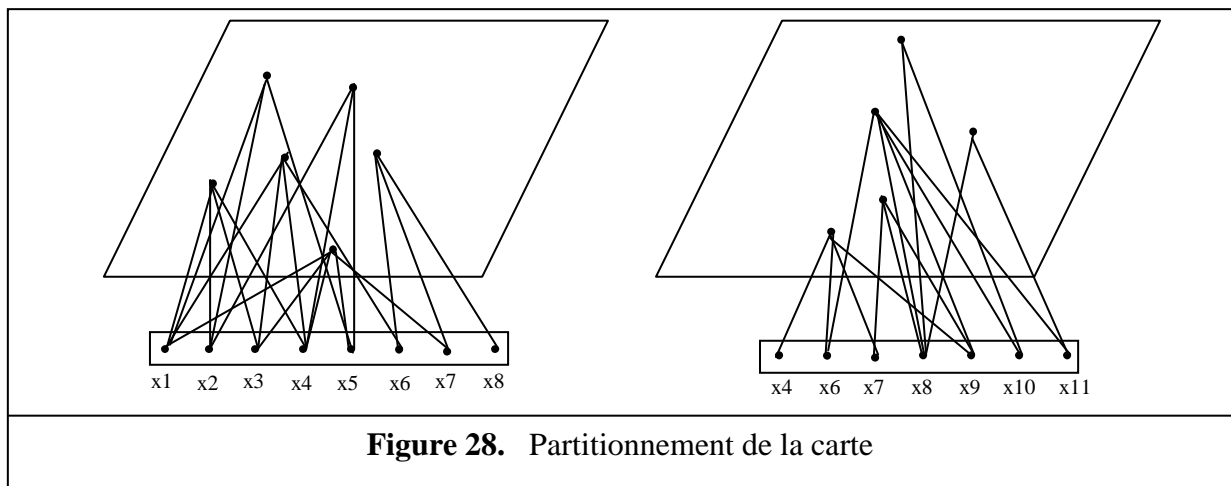
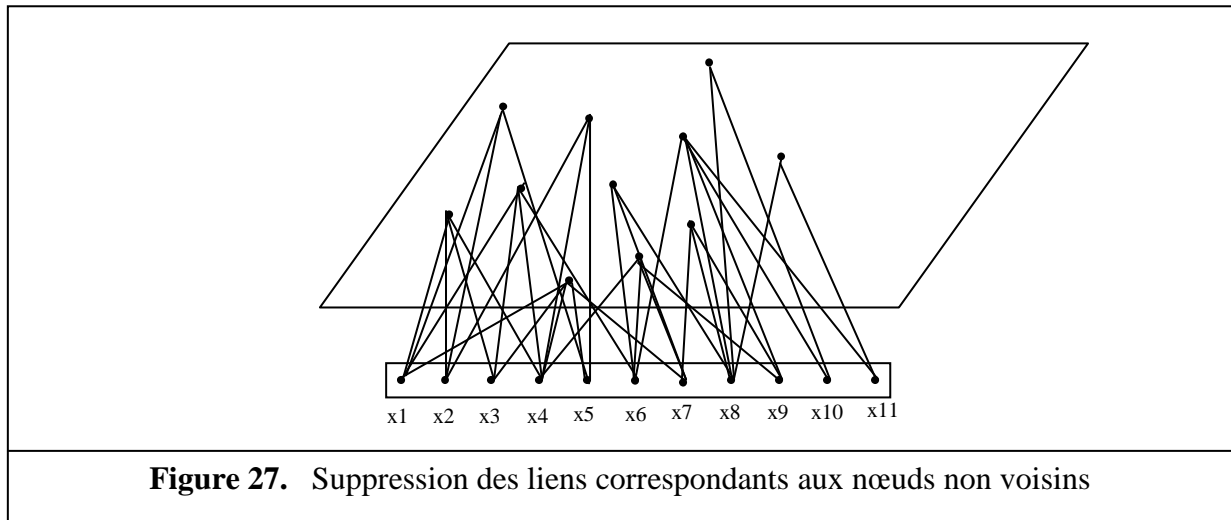
Comme T^{\min} (respectivement T^{\max}) représente la distance minimale (respectivement maximale) pour laquelle un nœud donné peut réagir, et comme il est proportionnel à la largeur du voisinage. Nous proposons de faire correspondre T^{\min} (respectivement T^{\max}) à δval^{\min} (respectivement δval^{\max}).

Ainsi, T^{\min} et T^{\max} doivent varier selon que le nombre de sensors change dans le temps (nœuds qui s'affaiblissent et qui meurent) et selon aussi que les données captées par les nœuds évoluent.

III.I.6 Répartition du traitement de clustering

Tel que montré sur la figure suivante, malgré que des nœuds soient proches entre eux géographiquement, les liens physiques entre eux ne sont pas forcément établis. C'est pour cette raison que nous proposons d'inclure la notion de voisinage des nœuds dans le processus de clustering.





x_4 , x_6 , x_7 , et x_8 peuvent appartenir à deux clusters différents du moment où ils font partie de deux espaces de compétition. Nous parlerons dans un prochain paragraphe dans ce document de ce genre de situations et comment que se fera l'affectation des rôles.

En effet, répartir le processus de clustering permet d'améliorer de manière considérable le temps de réponse de ce processus, ceci évitera d'échanger un nombre important de messages avec la station de base et rend le processus de clustering indépendant de cette dernière, ce qui est très utiles si le lien entre le réseau de capteurs et la station de base est perturbé. De plus, il permet d'alléger le traitement du clustering en fragmentant l'espace des observations.

III.II Principe de fonctionnement et fréquence d'exécution du clustering

Généralement, dans les protocoles de routage, le processus de clustering est lancé périodiquement après l'expiration des rounds. En effet, le déclenchement de ce processus ne doit se lancer que lorsque c'est nécessaire. Autrement dit, si après un certain intervalle de temps, la tête de cluster actuelle est toujours capable d'exécuter son rôle et reste dans la moyenne des autres nœuds concurrents, il n'y a pas nécessité de déclencher la procédure de clustering. A ce niveau là, plusieurs questions se posent par elles même :

III.II.1 Qu'est ce qui peut déclencher la restructuration des liens de clustering ?

Plusieurs facteurs peuvent être à l'origine de ce déclenchement, à savoir :

- Le niveau d'énergie de la tête de cluster actuelle ou alors son rapport $\frac{nv}{e}$ atteint un certain seuil. Ce seuil ne doit pas être une valeur fixe mais plutôt varier dans le temps en fonction de l'énergie restante au niveau des nœuds et du nombre de capteurs vivants dans le réseau (densité). Nous pouvons définir ce seuil à chaque calcul de clustering tel que pour chaque nœud r , nous calculons la valeur $\frac{nv_r}{e_r}$ et nous désignons une valeur moyenne de ces rapports comme seuil pour la période en question.
- Déplacement de la tête de cluster : lorsque le chef de groupe se déplace, un autre nœud doit le remplacer pour accomplir les tâches de routage.
- Tête de cluster défaillante : c'est dans le cas où le chef de groupe tombe en panne de manière inattendue (ex : désastre,...).

Changement considérable dans la structure du réseau. C'est lorsque beaucoup de nœuds se sont déplacés, sont épuisés, apparition d'obstacles entre les nœuds ...etc. Les capteurs vont essayer de se réassocier à leurs voisinages. Si une tête de cluster répond à cette demande, alors un lien sera construit entre les deux nœuds.

Lorsqu'une tête de cluster reçoit un nouvel élément dans son groupe, elle notifie son chef de groupe de ce changement. Lorsqu'un nombre important de changements est atteint (structure interne du groupe est modifiée), il serait temps de calculer le clustering pour ce super groupe.

III.II.2 Quelle est la portée du calcul du clustering ?

La nécessité du déclenchement du processus de regroupement n'implique pas forcément la restructuration de tout le réseau, il suffit de le faire pour le cluster lui-même ou à la limite aux groupes qui soient dans un proche voisinage.

III.II.3 Quelle fonction d'affectation χ utiliser ?

La fonction d'affectation permet d'associer chaque nœud (observation) à une tête de cluster (neurone vainqueur). Etant donné notre objectif est de répartir les charges sur l'ensemble des capteurs constituant le réseau, nous proposons ainsi d'utiliser l'expression suivante comme fonction d'affectation :

$$\chi^T(z) = \min_{\substack{r \in V \\ r \neq z}} (\delta val_{(r,z)}^T * \frac{nva_r}{e_r}) \quad (25)$$

tel que r n'est pas préalablement affecté à z et nva_r représente le nombre de voisins associés au nœud r . Initialement, sa valeur est égale à 1, car chaque nœud est responsable d'envoyer au moins ses données.

$\delta val_{(r,z)}^T$ exprime les différentes valeurs de $\delta val_{(r,z)}$ (δval entre le nœud r et les autres capteurs z) inférieures à T .

Quant à la condition que r ne soit pas préalablement associé à z , elle sera utilisée par la suite pour résoudre le problème des boucles.

III.II.4 Mise à jour des référents

A chaque itération et après avoir affecté les nœuds à leurs têtes de clusters, nous devons mettre à jour les vecteurs référents correspondants à chaque chef de groupe. Dans notre cas et comme nous n'utilisons pas la position géographique et la distance euclidienne, l'expression des référents peut être différente et représentée pour chaque nœud s par un vecteur de k

couplets, tel que k est le nombre de nœuds pour lequel le sensor s est chef de groupe. Les éléments des couplets (x,y) représentent respectivement l'identité du nœud appartenant au groupe et le poids reliant ce nœud en question (i) avec sa tête de cluster s .

Cette manière de faire nous permet de définir pour chaque tête de cluster, quels sont les nœuds qui appartiennent à son groupe. Alors que ça serait plus intéressant de connaître pour chaque nœud, quels sont les différents capteurs auxquels il peut s'associer et avec quel poids. Ainsi, lorsqu'une nouvelle exécution de clustering s'apprête à se lancer, les nœuds envoient leurs vecteurs de connexions au capteur calculateur du clustering. Ce dernier peut agréger ces vecteurs et les utiliser comme étant une initialisation intelligente des vecteurs référents dans le processus de regroupement.

$$w(1) = \left\{ \begin{array}{l} (2, p_{1,2}) \\ (4, p_{1,4}) \\ \dots \\ (k, p_{1,k}) \end{array} \right\} \text{ tel que } p_{i,j} \text{ représente le poids de connexion du nœud } i \text{ au capteur } j.$$

Par généralisation et après la phase d'affectation, nous gardons état de chaque affectation et nous construisons les vecteurs référents en utilisant la formule :

$$w(k) = \left\{ \begin{array}{l} (x, p_{k,x}) \\ (y, p_{k,y}) \\ \dots \\ (z, p_{k,z}) \end{array} \right\} \tag{26}$$

III.II.5 Peut-on avoir une boucle en exécutant l'algorithme d'auto-organisation ?

En effet, la condition ajoutée dans la formule de la fonction d'affectation ($r \rightarrow z$) nous permet d'éviter de tomber dans le cas des boucles dans un même cluster, c'est à dire le cas où deux nœuds soient chacun affecté au groupe de l'autre. Cette condition est forte et suffisante pour empêcher la formulation de liens bidirectionnels entre les capteurs d'un même groupe. Nous

étudierons dans la section **III.III.7** la résolution du problème des boucles dans le cas où des nœuds reçoivent plusieurs rôles de différents espaces de compétition.

La réassociation des nœuds qui sont CH à d'autres têtes de clusters peut aussi provoquer la construction des boucles. Pour remédier à ça, nous proposons de libérer les nœuds fils des têtes de cluster qui n'arrivent plus à communiquer avec le super CH. Ces nœuds vont à leur tour se réassocier s'ils sont de simples capteurs, sinon ils vont libérer à leur tour leurs nœuds fils et ainsi de suite.

III.II.6 Construction de la liste de voisinage ?

Afin d'établir la liste de voisinage, tout nœud envoie lorsque c'est nécessaire, un simple ping contenant son identité. Les capteurs recevant ce message insèrent cette nouvelle identité dans leurs listes de voisinage et ne sont pas obligés de répondre à ce ping. Ainsi, d'un côté, nous pouvons éviter l'échange de beaucoup de messages dans le réseau et d'un autre. Le nœud qui va calculer le clustering c'est lui qui aura une idée globale sur les sens des relations de voisinage (unidirectionnelle, bidirectionnelle).

III.II.7 Qui exécute le calcul du clustering ?

Les nœuds responsables de l'exécution de l'algorithme d'auto-organisation sont différents selon l'étape à laquelle le réseau se trouve, soit à l'initialisation du réseau ou après.

- A l'initialisation du réseau : initialement et afin de construire les liens de clustering, un nombre de messages doivent être échangés :
 - Chaque nœud envoie un message « I'm here » à son voisinage.
 - Chaque nœud va construire sa liste de voisins (l'ensemble de « I'm here » reçus).
 - Chaque nœud r publie son nombre de voisins nv_r .
 - Chaque nœud r envoie sa liste de voisins au nœud ayant une valeur maximale de nv parmi tous les nœuds adjacents tout en vérifiant que ce nœud choisi peut bien l'écouter aussi. Ces derniers vont calculer le clustering pour les nœuds qui leur ont envoyé leurs listes de voisins.

Ainsi, les capteurs ayant beaucoup de voisins peuvent calculer initialement le clustering pour un nombre important de nœuds et par conséquent, permettre d'avoir une vision plus globale sur la structuration du sous-ensemble du réseau.

- Après l'initialisation du réseau :
 - Quand la structure du réseau change de manière considérable, ceci s'explique par l'augmentation du nombre de notifications de déplacements des nœuds (réassociation), la tête de cluster actuelle lance le processus de calcul de clustering de son groupe seulement.
 - Lorsque la tête de cluster devient défaillante subitement, nous pouvons envisager de réassocier les capteurs, chacun à son prochain meilleur nœud sur la liste (sur le plan lien optimal). Ceci peut ne pas être une bonne solution, surtout si un nombre important de nœuds vont se réassocier à un même capteur. La solution dans ce cas là, consiste tout simplement à réassocier les nœuds à leur voisinage, ainsi, nous retombons dans le cas précédent où, la tête du super groupe s'occupera de relancer le clustering pour tout le super cluster lorsque le nombre de réassociations devient assez important. C'est à ce moment là, où les informations nécessaires aux initialisations des référents seront rapatriées en faisant appel à la notion des contextes.

Il reste à définir ce qui permet de détecter qu'une tête de groupe donnée est devenue défaillante. En effet, nous proposons d'utiliser les acquittements qui indiquent si les messages se transmettent correctement au CH. Si ce n'est pas le cas, il peut s'agir soit d'une défaillance de la tête de cluster, ou alors de la destruction du lien reliant un nœud donné à son chef de groupe. Dans les deux cas, le nœud va se réassocier et nous tomberons dans le cas précédent (nombre de réassociations important => calcul du clustering).

Ainsi, c'est à la responsabilité des nœuds simples de détecter que leur tête de groupe est défaillante, ceci nous évitera de lancer des processus au niveau des têtes de clusters qui vérifient en permanence que leurs fils sont toujours connectés à elles.

Remarque : Lorsque la tête du cluster x (groupe x) s'approche d'un certain seuil en capacité ou s'apprête à se déplacer, il n'est pas nécessaire de recalculer le clustering. Il suffit de libérer ses nœuds fils qui, à leur tour vont se réassocier à leurs nœuds voisins (même cas que pour la défaillance de la tête de cluster).

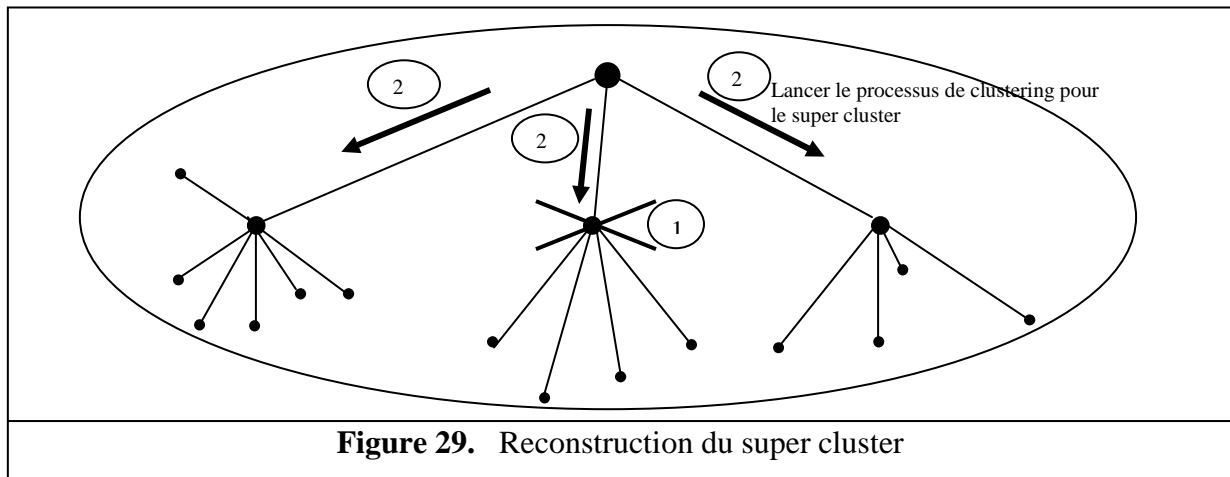


Figure 29. Reconstruction du super cluster

Si un nœud n'est attaché à aucune tête de cluster, il éteint son antenne et l'allume périodiquement afin de vérifier s'il y a pas eu de nouveaux voisins pour se réassocier, sinon il l'éteint encore pour préserver son énergie.

Lors du processus de clustering, un nœud peut recevoir plusieurs rôles. Prenons par exemple le cas montré dans le schéma du partitionnement de la carte (page 75). x_4 , x_6 , x_7 et x_8 appartiennent à deux espaces de compétition différents. Chacun de ces nœuds peut alors recevoir deux affectations issues de ces deux espaces. Ceci peut impliquer l'introduction des boucles. Pour remédier à ça, les têtes de clusters ne calculent le clustering que pour les nœuds qui leur envoient la liste de leurs voisins. Sachant qu'un nœud n'envoie la liste de ses voisins qu'au capteur qui a la valeur maximale de n_v dans son groupe.

III.III L'algorithme d'auto-organisation SOBC

Comme nous l'avons mentionné plus haut dans ce chapitre, nous proposons d'utiliser l'algorithme d'optimisation globale des cartes topologiques tout en utilisant la nouvelle fonction d'affectation définie dans un paragraphe précédent.

Self Organizing Based Clustering : SOBC

1. **Phase d'initialisation** : Effectuer l'algorithme nuées dynamiques des cartes auto-organisatrices pour la valeur $T=T_{\max}$, $t=0$ et jusqu'à stabilisation de la fonction de coût (tout en calculant la valeur de N_{iter}).
2. **Etape itérative t** : l'ensemble des référents W^{t-1} de l'étape précédente est connu, calculer la nouvelle valeur de T en appliquant la formule :

$$T = \delta val_{\max} * \left(\frac{\delta val_{\min}}{\delta val_{\max}} \right)^{\frac{t}{N_{iter}-1}}$$

Pour cette valeur du paramètre T, effectuer les deux phases suivantes :

- ❖ Phase d'affectation : On affecte chaque observation z_i aux référents définis à partir de l'expression (25) ;
- ❖ Phase de minimisation : Pour chaque nœud, appliquer la formule (26) afin de déterminer l'ensemble des nouveaux référents W^t

3. **Répéter** l'étape itérative jusqu'à ce que l'on atteigne $T = \delta val_{\min}$

IV Conclusion

Le clustering d'un réseau de capteurs reste une tâche assez complexe à réaliser, vu la densité de ses nœuds et la sensibilité des ondes radio qui constituent les liens entre les nœuds. Une approche monolithique s'avère inefficace dans ce genre d'environnements.

Nous avons exploité un modèle mathématique très puissant et utilisé dans le domaines de classification et d'auto-organisation. Les cartes SOM nous permettent d'organiser le réseau en clusters tout en se basant sur l'optimisation d'un paramètre et ainsi assurer la survivabilité maximale du réseau.

L'application de ces cartes ne peut être faite sans adaptation de ses fonctions et paramètres. Cette adaptation dépend des caractéristiques des nœuds (limitation énergétique) et des liens les reliant.

Nous avons évité le maximum de traitements possibles tel que nous évitons tout calcul inutile (des calculs périodiques non nécessairement utiles, des sous ensembles du réseau non concernés par le changement d'une structure du réseau,...).

Nous avons essayé de trouver des critères plus adéquats pour le choix des têtes de clusters qui permettent d'établir un compromis entre le niveau en énergie et le nombre de voisins.

Quatrième chapitre

Quatrième chapitre : **Simulations et évaluation de performances**

Simulations et évaluation de performances

Afin de pouvoir mesurer les performances de notre solution, nous proposons d'utiliser le même environnement qui a été considéré dans les travaux de LEACH et LEA2C, à savoir :

- Le réseau se compose de 100 nœuds homogènes avec pour énergie $e=2j$ et dispersés aléatoirement dans un espace et qui peuvent être mobiles.
- La station de base est positionnée dans un endroit assez distant du reste du réseau.
Remarque : la superficie du réseau et les coordonnées des différents nœuds nous intéressent pas dans nos simulations, vu que notre protocole n'est pas géographique. Nous nous basons plutôt sur les valeurs des données captées afin de définir le rapprochement qui existe entre les nœuds.
- Le modèle de communication étant le même utilisé dans LEACH et LEA2C.

Nous avons développé et simulé notre protocole sous TinyOs 2.x qui constitue un environnement complet de développement pour les réseaux de capteur sans fil. Il inclut un système d'exploitation open-source conçu pour les systèmes embarqués (plus particulièrement pour les WSN), et presque toutes les firmes de production de capteurs testent leurs plateformes avec ce système d'exploitation qui est devenu la solution de facto du marché.

Nous avons réussi à construire un algorithme de clustering hiérarchique à plusieurs niveaux. Ceci nous a permis d'assurer au mieux la connectivité du réseau avec la station de base. SOBC construit de proche en proche les liens reliant les nœuds entre eux et le réseau à la station de base.

I Les métriques

Dans notre travail, nous nous intéressons particulièrement à l'optimisation de la consommation d'énergie. Par conséquent, l'énergie constitue la métrique principale à calculer, plus certains paramètres qui lui sont fortement liés. Nous proposons alors de mesurer les métriques suivantes :

- Fréquence d'exécution de la procédure de clustering.
- Le nombre de messages échangés pour le processus de clustering.
- Fiabilité du clustering.
- Energie consommée et durée globale du réseau.
- Durée de vie des nœuds par rapport à la durée de vie globale du réseau.

I.I Fréquence d'exécution du clustering

La fréquence d'exécution du clustering constitue un paramètre clé dans notre solution afin de minimiser la consommation d'énergie.

Nous avons réussi à réduire le nombre moyen de la participation des nœuds à la procédure de clustering de manière très considérable, tel que comparant à LEA2C dans sa variante qui choisit le nœud ayant le maximum d'énergie comme tête de cluster.

Nous avons réduits cette participation à 96% tel que nous avons considéré que les nœuds sont programmés à capter et à envoyer des données chaque 10 minutes. Ainsi, dans LEA2C, la station de base recalcule le clustering et renvoie les rôles aux capteurs du réseau toutes les 10 minutes.

Quant à LEACH, la valeur de l'intervalle de réélection des têtes de clusters n'est pas définie. Dans notre solution, le nombre moyen de participations d'un nœud donné au processus de clustering dans un environnement stationnaire ne dépasse pas $\frac{1}{25}$ le nombre de participations dans LEA2C.

I.II Le nombre de messages transmis pour le clustering

Bien que nous utilisons plusieurs messages afin d'obtenir les données nécessaires au clustering (liste des voisins), mais le nombre global de ces messages reste inférieur à celui de LEA2C et LEACH qui exécutent leurs algorithmes de clustering de manière fréquente.

Si nous considérons le réseau à son état initial, nous pouvons comparer le nombre de messages échangés entre notre solution et le protocole LEA2C. Supposons que nous avons k têtes de clusters et n nœuds simples, nous aurons les messages suivants à échanger :

SOBC

- k messages de broadcast
 - n réceptions par les nœuds
 - n pings
 - m réceptions par les noeuds
 - n envois de listes de voisins aux CH
 - k réceptions par les CH
- calcul du clustering par la CH
- n envois de rôles vers les noeuds
 - n réceptions par les nœuds

$$\text{nb_msg} = 2n*\text{réceptions} + 3n*\text{envois} + k*\text{envois} + k*\text{réceptions} + m*\text{réceptions}$$

LEA2C dans un cas stationnaire

- k réceptions par les CH
- n envois vers les nœuds
- n réceptions par les nœuds

$$\text{nb_msg} = k*\text{réceptions} + n*\text{envois} + n*\text{réceptions}$$

LEA2C dans un cas non stationnaire

- n envois de coordonnées au CH
 - n réceptions par les CH
 - k envois vers la SB
- calcul du clustering par la SB
- k réceptions par les CH
 - n envois vers les nœuds
 - n réceptions par les nœuds

$$\text{nb_msg} = 2n*\text{réceptions} + 2n*\text{envois} + k*\text{envois} + k*\text{réceptions}$$

m étant le nombre de messages reçus après diffusion des pings, nous ne pouvons pas définir sa valeur car elle est variable et dépend des positions géographiques, des liens physiques reliant les différents nœuds et aussi du nombre de capteurs vivants dans le réseau. Mais nous avons une estimation qui dit que m est souvent supérieur à n.

D'après le nombre de messages échangés, nous constatons la différence marquée dans le nombre des messages transmis pour l'algorithme de clustering. Cependant, dans le protocole LEA2C, si le critère de choix de la tête de cluster est le nœud ayant le niveau maximum d'énergie, le calcul du clustering se fait après chaque transmission. Par contre dans notre protocole, le calcul de la structure se fait souvent pour un groupe seulement et non pas pour la totalité du réseau. Ceci fait qu'au bout d'une certaine période, cette différence sera rattrapée.

Comme notre solution s'exécute de la même manière dans le cas stationnaire et non stationnaire, nous comparons le nombre moyen de ces messages échangés avec celui de LEA2C dans le cas non stationnaire.

Sachant que pour une exécution, nous avons une différence de « $n \cdot \text{envois} + m \cdot \text{réceptions}$ ». Et puisque le nombre moyen global de participations au clustering dans notre solution est de $\frac{1}{25}$ le nombre de participations dans LEA2C. Donc un nœud dans LEA2C va participer en moyenne 24 fois à la procédure de clustering, et à la 25^{ième}, on aura une exécution parallèle dans SOBC. Ce qui donne une différence de :

$$24 \cdot (2n \cdot \text{réceptions} + 2n \cdot \text{envois} + k \cdot \text{envois} + k \cdot \text{réceptions}) - (n \cdot \text{envois} + m \cdot \text{réceptions})$$

C'est là où nous ressentons l'utilité de se baser sur le paramètre de fréquence d'exécution du clustering. C'est ce qui nous a permis aussi de gagner en consommation d'énergie.

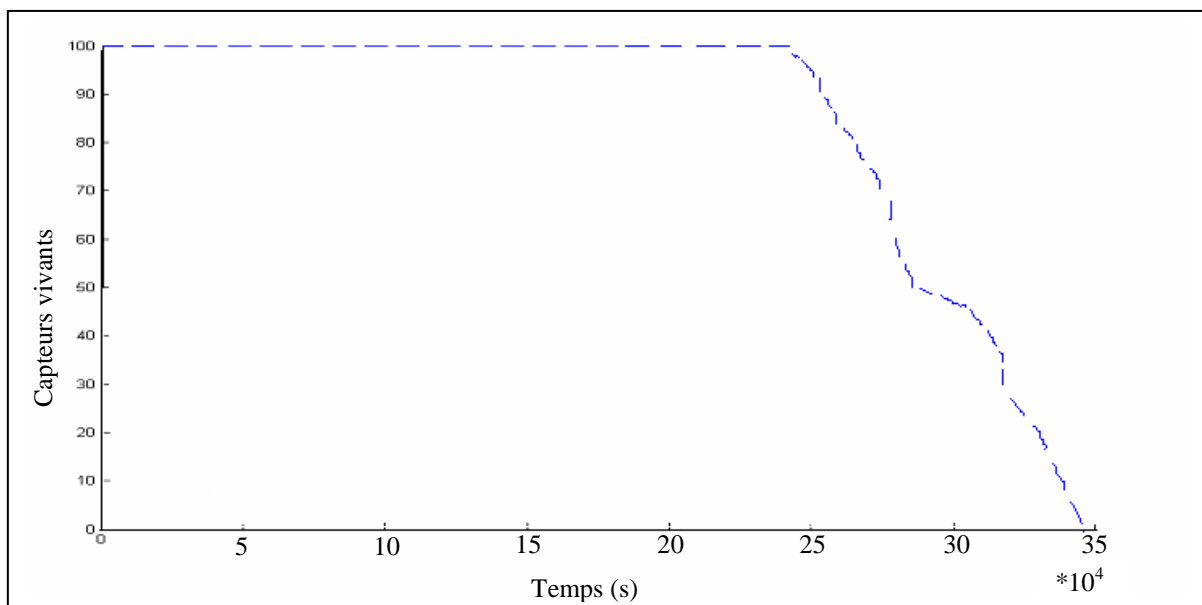
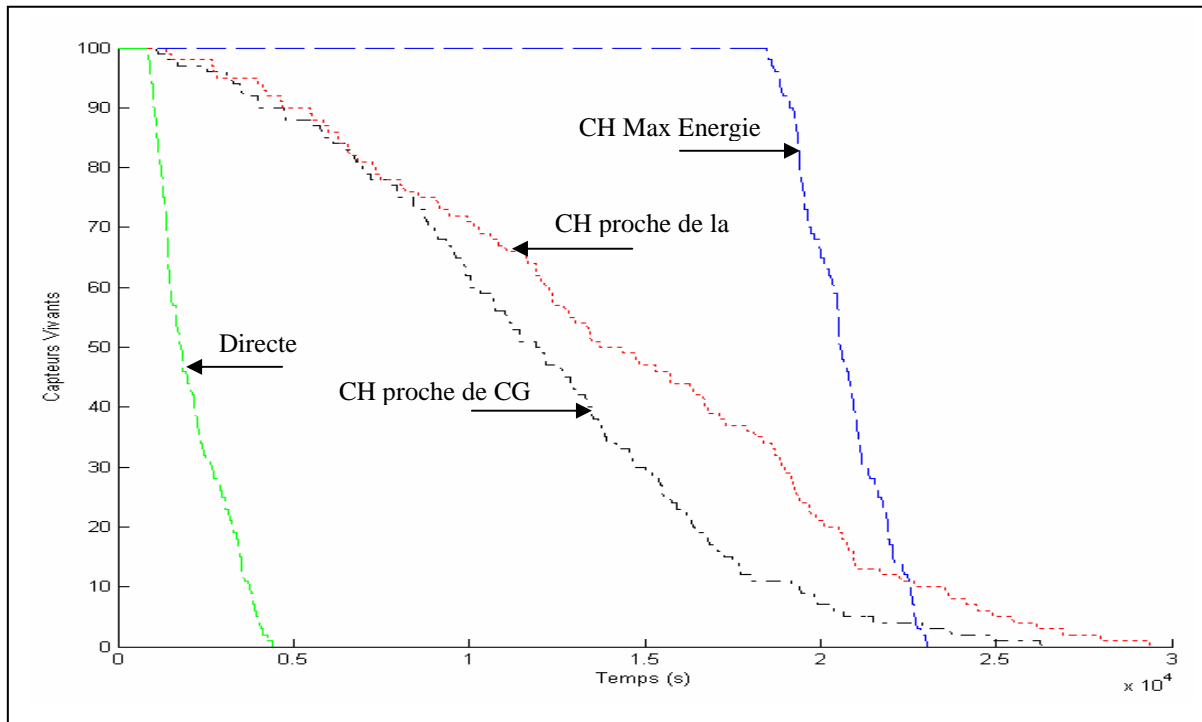
I.III Concordance du clustering résultant avec les liens physiques réels

Ceci est un paramètre très important et déterminant dans le bon fonctionnement du réseau, tel que si un nœud est affecté à une tête de cluster qui ne peut pas l'écouter, ses données seront perdues et il consommera son énergie inutilement.

Nous avons pu réaliser une affectation efficace dans 98% des cas, grâce à la prise en compte de la notion de voisinage dans le processus de clustering. Les 2% restants, représentent les cas où les liens entre deux nœuds se perturbent de manière inattendue, ce qui est dû à la nature des ondes radio qui restent très fragiles aux différents phénomènes climatiques et physiques.

Nous avons marqué un gain considérable par rapport à LEA2C qui se base sur la position géographique pour déterminer les relations de clustering et qui n'a pas considéré les liens réels entre les nœuds. Dans 50% des cas simulés, les nœuds sont affectés à des têtes de clusters qui ne sont pas dans leur écoute, ou des chefs de groupes qui envoient des rôles à des nœuds fils qui ne peuvent pas les écouter.

I.IV Energie consommée et durée de vie des nœuds par rapport à la durée de vie globale du réseau



Sur les graphes précédents, nous constatons l'amélioration de la durée de vie globale du réseau qui est due à la diminution de la consommation d'énergie. En terme de cette durée de

vie, notre solution offre un gain de 12 fois plus important que dans le protocole LEA2C avec le meilleur critère de choix des CHs.

Quant à la survivabilité des capteurs dans le réseau, nous avons constaté que 100% des nœuds restent vivant pendant 70% de la durée de vie globale du réseau.

II Conclusion

Nous avons réalisé un algorithme de routage à plusieurs niveaux d'hierarchie complètement distribué et qui construit les liens entre les nœuds de manière très fiable tenant compte des liens de voisinage réels entre ces capteurs.

Nous avons pu concrétiser une optimisation importante dans la consommation d'énergie grâce à l'utilisation de l'intelligence de l'apprentissage évolutif et non supervisé des cartes SOM.

Les résultats obtenus sont très prometteurs et encourageants, tel que la durée de vie du réseau a été nettement améliorée avec un taux très satisfaisant de survivabilité de la majorité des nœuds par rapport à la durée de vie globale du réseau.

Conclusion générale

Conclusion générale

Le domaine des réseaux de capteurs est une technologie qui connaît un succès sans précédent dans le domaine des environnements intelligents. Ce succès est issu de l'évolution croissante et continue de la technologie de communication sans fil et celle des réseaux mobiles.

Les réseaux de capteurs sont des réseaux Ad hoc mais à quelques spécificités près, dont la plus déterminante, la limitation en énergie.

En effet, toutes les solutions proposées pour les réseaux de capteurs doivent prendre en charge comme priorité absolue l'économie en consommation d'énergie, entre autres les protocoles de routage.

Nous pouvons distinguer deux techniques différentes de routage qui sont soit à base de sauts ou alors hiérarchiques. Ceci est une classification très globale tel que nous pouvons trouver d'autres classifications dans la littérature, chacune se basant sur une métrique bien déterminée, nous donnons à titre d'exemple la classe des protocoles de routage multiples qui visent à améliorer la tolérance aux pannes. Comme nous pouvons trouver des protocoles hybrides qui appartiennent à plusieurs classes.

Le routage à base de sauts consiste à envoyer les paquets dans le voisinage de la source de données, les nœuds se trouvant dans le chemin menant vers le puit jouent le rôle de routeurs et acheminent les paquets à leur tour vers leur voisinage et ainsi les messages sont transmis de proche en proche jusqu'à atteindre le puit.

Plusieurs classes de protocoles appartiennent à cette technique de routage parmi lesquelles nous distinguons :

- Les protocoles centrés données qui utilisent des schémas de définition et d'identification de données.
- Les protocoles géographiques pour lesquels le paramètre de localisation joue un rôle important dans la détermination du chemin à emprunter.
- Les protocoles orientés qualité de services qui essaient d'établir un compromis entre la consommation de ressources et la qualité de données livrées.

Conclusion générale

Le routage hiérarchique se base sur la construction de groupes (clusters), chaque groupe possédant une tête de cluster qui a pour rôle, d'agréger les données reçues par les nœuds membres du groupe et de les acheminer vers le puit. Cette technique vise la scalabilité, elle a offert un apport positif permettant de réduire et de manière très significative, la consommation d'énergie et ce en éliminant un nombre très important de transmissions entre tous les nœuds et la station de base. Beaucoup de recherches ont été concentrées sur cette approche hiérarchique vu les gains prometteurs issus des protocoles proposés dans cette classe et pour lesquels, les résultats de simulation ont prouvé de meilleures performances surtout pour la prolongation de la durée de vie du réseau global.

La phase de construction des liens hiérarchiques reste coûteuse en énergie car un nombre très important de messages sont échangés afin de construire et de maintenir ces groupes.

Les cartes auto-organisatrices (SOM) constituent une méthode de clustering optimale basée sur divers algorithmes dont le plus connu, l'algorithme de Kohonen. Elles permettent de définir pour un grand nombre de données de grandes dimensions, un groupe de référents résumant chacun, un sous ensemble de ces données. De même que dans les réseaux de capteurs où la tête de cluster doit agréger les données issues de son groupe. Généralement, ces données sont homogènes vu que les nœuds sont denses, alors la probabilité que ces nœuds captent des données proches en valeurs est assez grande.

Nous avons exploité les cartes SOM dans la procédure de regroupement des réseaux de capteurs quant à la convergence de leurs objectifs à savoir le clustering. Cette tendance nous a permis de baser cette procédure sur un algorithme mathématique qui assure la convergence de son exécution vers un regroupement optimal par rapport à un paramètre donné qui, dans notre cas, est la préservation de la consommation d'énergie. Nous avons exprimé ce paramètre en fonction du voisinage et de la variation des données.

L'exploitation distribuée de ces cartes nous a permis d'améliorer le temps de réponse de la procédure de calcul de clustering et la qualité de l'information retournée par le réseau sensor, ainsi que de prendre en considération les relations de voisinage réelles entre les nœuds.

Conclusion générale

Nous avons réalisé un algorithme de routage à plusieurs niveaux d'hierarchie complètement distribué et qui construit les liens entre les nœuds de manière très fiable tenant compte des liens de voisinage réels entre ces capteurs.

Les simulations ont montré que la durée de vie du réseau a été nettement améliorée avec un taux très satisfaisant de la vie de la majorité des nœuds par rapport à la durée de vie globale du réseau.

Nous avons considéré le facteur de fréquence d'exécution du clustering, ce paramètre est vraiment déterminant dans l'optimisation des ressources énergétiques.

Perspectives :

Ces résultats satisfaisants nous encouragent à continuer nos recherches dans ce domaine afin d'améliorer notre solution. Nous proposons comme perspectives à notre travail :

- Etudier la définition optimale du paramètre « seuil » dans l'algorithme SOM ainsi que sa loi de décroissance.
- Analyser et améliorer la qualité des données agrégées et retournées par notre algorithme pour les différentes fonctions d'agrégation.
- Utiliser d'autres versions de cartes SOM.

Bibliographie

Bibliographie

[Akyildiz et al. 02]: «Wireless Sensor Networks: a survey» I.F.Akyildiz, W.Su, Y.Sankarasubramaniam, E.Cayirci. Computer Networks: The International Journal of Computer and Telecommunication Networking, v.38 n.4, pp.393-422, 2002.

[Al-karabi and al. 04]: « Data Agregation in Wireless Sensor Networks – Exact and Approximate Algorithms » J. Al-Karabi, R.UI-Mustapha, A.E.Kama, in Proceedings of IEEE Workshop on High Performance Switching and Routing (HPSR), April 18-21, 2004, Phoenix, Arizona, USA.

[Al-karabi, Kama 03] : « Routing Techniques in Wireless Sensor Networks :A Survey » J.Al-Karabi, A.E.Kama, Dept. of Electrical and Computer Engineering, Iowa State University, Ames, Iowa 50011, 2003.

[Anouar et al. 97]: « Self Organized Map, A Probabilistics Approach », Anouar F., Badran F., Thiria S. Proceedings of the workshop on Self-Organized Maps, Helsingy University of Technology, Espoo, Finlande [4-6 Juin 1997]

[Bakre, Badrinath 95] : « I-TCP : indirect TCP for mobile hosts » A.Bakre, B.R.Badrinath, Proceedings of the 15th International Conference on Distributed Computing Systems, Vancouver, BC, pp.136 –143, May 1995 .

[Braginsky, Estrin 02] : « Rumor Routing Algorithm for Sensor Networks » D. Braginsky and D.Estrin in the Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA, October 2002.

[Bus 99] : « 21 ideas for the 21st century», Business Week, pp.78-167, august. 30 1999.

- [Dehni et al.05] : « Power Control and Clustering in Wireless Sensor Networks » L.Dehni, F.Krief, Y.Bennani 2005.
- [Dreyfus et al. 04] : « Réseaux de neurones, méthodologie et applications», Dreyfus G., Martinez J.-M., Samuelides M., Cordon M.B., Badran F., Thiria S., Hérault L. 2 édition, 7, p.295-356 [2004]
- [Duda et al. 73]: « Pattern Classification and Scene Analysis », Duda R.O., Hart P.E. John Wiley [1973].
- [Dulman and al. 03]: « Trade-Off between Traffic Overhead and Reliability in Multipath Routing for Wireless Sensor Networks » S.Dulman, T.Nieberg, J.Wu, P.Havinga, WCNC Workshop, New Orleans, Louisiana, USA, March 2003.
- [Fleury, Simplot-Ryl 04]: «Réseaux de capteurs» E.Fleury, David Simplot-Ryl, CITI/INSA de Lyon ARES/INRIA.
<http://www.lri.fr/~fragile/IMG/pdf/ACISecFragileSensorEFL-P1.pdf>
- [Gaul et al. 00]: « Data Analysis Scientific Modeling and Practical Application», Gaul W., Opitz O. Schader M. Springer [2000].
- [He and al.03]: «SPEED: A stateless protocol for real-time communication in sensor networks» T.He and al., in the Proceedings of International Conference on Distributed Computing Systems, Providence, RI, May 2003.
- [Heinzelman et al. 00] : « Energy-efficient communication protocol for wireless micro sensor networks » W.Heinzelman, A.Chandrakasan, H.Balakrishnan. In Proceedings of the Hawaii Conference on System Sciences, Jan. 2000.
- [Heinzelman et al. 99] : « Adaptive Protocols for Information Dissemination in Wireless Sensor Networks » W.Heinzelman, J.Kulik, H.Balakrishnan. Proc.5th

ACM/IEEE Mobicom Conference (Mobicom '99), Seattle, WA, August, 1999. pp.174-85.

[Ibriq 04] : « Cluster based routing in wireless sensor networks, issues and challenges »
J.Ibriq, I.Mahgoub. SPECT 2004.

[IEEE 802.3 Working Group]: « IEEE 802.3 CSMA/CD (ETHERNET) »
<http://grouper.ieee.org/groups/802/3/>

[Intanagonwiwat 03]: « Directed Diffusion for Wireless Sensor Networking » C. Intanagonwiwat, R.Govindan, D.Estrin, J.Heidemann and F.Silva. IEEE/ACM TRANSACTIONS ON NETWORKING, VOL.11, NO.1, february 2003.

[Jain et al. 88]: « Algorithms for Clustering Data », Jain A.K., Dubes R.C. Prentice Hall [1988]

[Johnson and al. 96]: « Dynamic Source Routing in Ad Hoc Wireless Networks » D.B.Johnson and al., in Mobile Computing, edited by Tomas Imielinski and Hank Korth, Kluwer Academic Publishers, ISBN: 0792396979, 1996, Chapter 5, pages 153-181.

[Karlof, Wagner 03]: « Secure routing in wireless sensor networks: attacks and countermeasures » C.Karlof and D.Wagner, Ad Hoc Networks, Vol.1 (2003) pp.293315.

[Karp, Kung 00]: «GPSR: Greedy perimeter stateless routing for wireless sensor networks » B.Karp and H.T.Kung, in the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00), Boston, MA, August 2000.

[Khelladi, Badache 04]: «Les réseaux de capteurs: Etat de l'art» L.Khelladi, N.Badache. Rapport de mini-projet. N° LSI-TR0304 Février 2004.

[Kohonen 01]: « Self Organizing Maps », Kohonen T. Springer, 3 édition [2001]

- [Kulik et al. 02] : « Negotiation-based protocols for disseminating information in wireless sensor networks » J.Kulik, W.R.Heinzelmen and H.Balakrishnan. Wireless Networks, Volume:8, pp.169-285, 2002.
- [Lindsey 02]: «PEGASIS, Power-Efficient Gathering in Sensor Information Systems» S. Lindsey, C.Raghavendra: IEEE Aerospace Conference Proceedings, 2002, Vol.3,9-16 pp.1125-1130.
- [Luttrel 94]: « A bayesian analysis of Self Organizing Maps», Luttrel S.P. Neural Comput, 6[1994]
- [Manjeshwar 01] : «TEEN : a routing protocol for enhanced efficiency in wireless sensor networks» A. Manjeshwar and D.P.Agarwal: In 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, April 2001.
- [Manjeshwar 02] : «APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks » A. Manjeshwar and D.P.Agarwal. Parallel and Distributed Processing Symposium, Proceedings International, IPDPS 2002, pp.195-202.
- [McCulloch et al. 43]: «A logical calculus of the ideas immanent in nervous activity», McCulloch W.S., Pitts W. Bulletin of mathematical Biophysics, 5, p.159-200 [1944]
- [Minsky et al. 69]: « Perceptrons », Minsky M, Papert S. MIT Press [1969]
- [Oja et al. 99]: « Kohonen Maps », Oja E., Kaski S. Elsevier [1999]
- [Perkins and al. 2002]: «Ad hoc On-Demand Distance Vector (AODV) Routing » C.Perkins and al., Internet Draft draftietf-manet-aodv-11.txt, June 2002 (work in progress).

- [Rentala et al.01] : « Survey on Sensor Networks » P.Rentala, R.Musunnuri, S.Gandham, U.Saxena, Proc. of International Conference on Mobile Computing and Networking, University of Texas Dallas 2001.
- [Saporta 90] : « Probabilités, analyse des données et statistiques », Saporta G. Technip [1990]
- [Savvides 01] : « Dynamic Fine-grained localization in Ad hoc Networks of sensors » A.Savvides, C-C Han, and M.Srivastava: Proceedings of the seventh ACM Annual International Conference on Mobile Computing and Networking (MobiCom), July 2001. pp.166-179.
- [Schurgers, Srivastava 01]: « Energy efficient routing in wireless sensor networks » C. Schurgers and M.B.Srivastava, in the MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force, McLean, VA, 2001.
- [Shal and Rabaey 02]: « Energy Aware Routing for Low Energy Ad Hoc Sensor Networks » R.Shal and J.Rabaey, in the Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Orlando, FL, March 2002.
- [SILVA et al. 04] : « An overview of directed diffusion» F.Silva, J.Heidemann, R.Govindan, D.Estrein 2004.
- [Subramanian 00]: « An architecture for Building Self Configurable Systems». L. Subramanian and R.H.Katz: in the proceedings of IEEE / ACM Workshop on Mobile Ad hoc Networking and Computing, Boston, MA, August 2000.
- [Tian 03] : « A Scalable Sensor Network Architecture for Robustness and Time Energy Awareness » H.Tian. Department of Computer Science, University of Virginia. Olsson 263D, April 2003.
- [XU et al. 01] : « Geography-informed Energy Conservation for Ad hoc Routing » Y.Xu, J.Heidemann and D.Estrin. In Proceedings of the seventh annual

Bibliographie

ACM/IEEE International Conference of Mobile Computing and Networking 2001, pp.70-84.

[Yu et al. 01]: «Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks» Y.Yu, D.Estrin and R.Govindan. UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, May 2001.

[Zeghilet 05] : «Le routage dans les réseaux de capteurs » H.Zeghilet : Rapport de mini-projet USTHB 2005.