

TABLE DE MATIERE

CHAPITRE I. INTRODUCTION GENERALE.....	1
CHAPITRE II. SYSTEMES SENSIBLES AU CONTEXTE	4
II.1. Introduction	4
II.2. Notion De Contexte	5
II.2.1. Notion De Contexte Dans Le Domaine De l'informatique Pervasif.....	5
II.2.2. Catégories de Contexte.....	6
II.2.3. Qualité du contexte.....	7
II.2.4. Utilisations du Contexte dans les Systèmes Informatiques	7
II.2.5. Caractéristiques de L'information de Contexte.....	8
II.2.6. Délivrance du Contexte à l'Application	8
II.2.7. Sources d'informations.....	9
II.3. Notion de Sensibilité au Contexte.....	9
II.3.1. Classes de Sensibilité au Contexte	10
II.3.2. Architecture d'un Système Sensible au Contexte	12
II.3.3. Modèles de Contexte	13
II.3.4. Conception des Applications Sensibles au Contexte.....	14
II.4. Conclusion	15
CHAPITRE III. RESEAUX DE CAPTEURS SANS FIL	16
III.1. Introduction	16
III.2. Réseaux Sans Fils	16
III.2.1. Définition	16
III.2.2. Classification des Réseaux Sans Fil.....	16

III.3. Réseaux de Capteurs Sans Fil.....	21
III.3.1. Capteurs Sans Fil.....	21
III.3.2. Description des réseaux de capteurs sans fil	24
III.3.3. Technologies de la communication dans les réseaux de capteurs sans fil	28
III.3.4. Types de réseaux de capteurs sans fil	28
III.3.5. Objectif de base du réseau de capteurs sans fil	29
III.3.6. Caractéristiques des réseaux de capteurs sans fil	29
III.3.7. Défis d'un système de réseau de capteurs sans fil	31
III.3.8. Classification des applications des réseaux de capteurs sans fil	31
III.4. Relation entre RCSF et les systèmes sensibles au contexte	32
III.5. Conclusion	34
CHAPITRE IV. TRAVAUX CONNEXES.....	35
IV.1. Introduction	35
IV.2. Classification des Systèmes Sensibles au Contexte	35
IV.2.1. Réseaux de Capteurs Sensibles au Contexte	35
IV.2.2. Systèmes Sensibles au Contexte à base des Réseaux de Capteurs.....	35
IV.3. Exemples de Travaux de chaque Classe.....	36
IV.3.1. Réseaux de Capteurs sensibles au contexte	36
IV.3.1.1. A Context-Aware Approach to Conserving Energy in Wireless Sensor Networks	36
IV.3.1.2. Context-Aware Sensors	36
IV.3.1.3. Context-Aware Sensornet.....	36
IV.3.2. Sensibles au Contexte à base des Réseaux de Capteurs.....	36
IV.3.2.1. Enhanced sensor network:	36
IV.3.2.2. SAIL.....	37

IV.3.2.3. CAS ²	38
IV.3.2.4. A Context Aware Motor Controller Based on WSN	38
IV.3.2.5. CONSORTS-S	39
IV.3.2.6. Context-aware physiological data acquisition and processing with wireless sensor networks.	40
IV.3.2.7. Study on the Context-Aware Middleware for Ubiquitous Greenhouses Using Wireless Sensor Networks.....	41
IV.3.2.8. Using Wireless Sensor Networks to Narrow the Gap between Low-Level Information and Context Awareness.....	42
IV.4. Conclusion	43
CHAPITRE V. CONTRIBUTION.....	44
V.1. Introduction	44
V.2. Partie 1 : La tolérance aux Pannes dans les RCSF	44
V.2.1. Panne	44
V.2.1.1. Définition : Faute, erreur et faille	44
V.2.1.2. Causes des Pannes dans les RCSF	45
V.2.1.3. Classification des Pannes	45
V.2.2. Tolérance aux Pannes	46
V.2.3. Classification des solutions de tolérance aux pannes dans les RCSF ..	48
V.2.3.1. Classification selon la détection d'erreurs	48
V.2.3.2. Classification selon la correction d'erreurs	49
V.3. Partie 2 : Les Travaux de base de mon projet.....	52
V.3.1. Travail 1 : Système sensible au contexte à base du réseau de capteurs sans fil.....	52
V.3.1.1. Introduction.....	52
V.3.1.2. Problématique du Sujet de Recherche	52
V.3.1.3. Concept de Produit Actif	52

V.3.1.4.	Modèle de Produit Actif Proposé par Zouinkhi.....	53
V.3.1.5.	Tâches du Produit Actif	55
V.3.1.6.	Comportement du Produit Actif	55
V.3.1.7.	Scénario de Fonctionnement Complet du Réseau	56
V.3.1.8.	Messages Echangés dans le Réseau.....	57
V.3.1.9.	Conclusion et Critique	57
V.3.2.	Travail 2 : Détection de pannes dans les RCSF	58
V.3.2.1.	Introduction.....	58
V.3.2.2.	Approche Bayesian	58
V.3.2.3.	Principe de détection de fautes	58
V.3.2.4.	Conclusion et Critique	58
V.3.3.	Travail 3: Détection et correction de pannes dans les RCSF.....	58
V.3.3.1.	Introduction.....	58
V.3.3.2.	Technique Proposée	59
V.3.3.3.	Types de Pannes Traitées.....	59
V.3.3.4.	Aperçu de la technique	59
V.3.3.5.	Conclusion	62
V.4.	Partie 3 : la Contribution	63
V.4.1.	Notre Objectif.....	63
V.4.2.	Description de la Structure du Réseau.....	64
V.4.3.	Fonctions de Chaque Composant du Réseau.....	65
V.4.4.	Caractéristiques du réseau	65
V.4.5.	Caractéristiques du Maître de la Zone	65
V.4.6.	Caractéristiques du Produit Actif	66
V.4.7.	Fonctionnement de l'Application	67

V.4.8. Processus de Chaque Fonction dans l'Application	68
V.4.8.2. Division du Réseau en Quatre Zones	69
V.4.8.3. Détermination des Maîtres des Zones	71
V.4.8.4. Surveillance de L'environnement par les Produits	72
V.4.8.5. Recherche de la Panne et sa Correction	73
V.4.8.5.1. Types des Pannes à Rechercher	73
V.4.8.5.2. Processus de recherche de panne	74
V.4.8.5.3. Détermination du Nœud en Panne	75
V.4.8.6. Recherche de la Panne de sensation	76
V.5. Conclusion	78
CHAPITRE VI. IMPLIMENTATION ET SIMULATION	79
VI.1. Introduction	79
VI.2. Simulateurs de Réseau Existants	79
VI.3. Raison de Choisir le Simulateur CASTALIA	80
VI.4. Description du Simulateur CASTALIA	80
VI.5. Simulation.....	82
VI.5.1. Environnement de Simulation.....	82
VI.5.2. Paramètres Utilisés dans la Simulation	83
VI.6. Résultats de Simulation	83
VI.6.1. Salutation entre les Produits Actifs	83
VI.6.2. Vérification de la Compatibilité.....	84
VI.6.3. Cas du Déclenchement d'Alerte.....	85
VI.6.4. Détection du Nœud Défectueux	86
VI.6.5. Confirmation de la panne	86
VI.6.6. Correction de la panne	87

VI.7. Conclusion	88
CONCLUSION GENERALE.....	89
REFERENCES	90



LISTE DES TABLEAUX

Tableau III-1 Exemples des protocoles de la couche réseaux	26
Tableau III-2 Les défis d'un système réseaux de capteur	31
Tableau V-1 base des connaissances du produit actif.....	54
Tableau V-2 Ensemble des messages échangés requête/réponses.....	57
Tableau V-3 Table des informations.....	60
Tableau V-4 Table des enregistrements.....	61
Tableau V-5 table grille	61
Tableau V-6 Fonctions des composants du réseau.	65
Tableau V-7 Table d'enregistrement	66
Tableau V-8 Table d'information	66
Tableau V-9 Table grille.....	66
Tableau V-10 base de connaissance du produit.....	67
Tableau V-11 Table des produits actifs(PAs).....	69
Tableau V-12 Table des zones avec leurs maîtres et nœuds.....	71

LISTE DES FIGURES

Figure I-1 Résumé Des Idées à Réaliser.	2
Figure II-1 Évolution vers l'informatique diffuse.	4
Figure II-2 Architecture générale d'un système sensible au contexte.	12
Figure III-1 Classification des réseaux sans fil.....	17
Figure III-2 Classification des réseaux sans fil suivant leur taille.	17
Figure III-3 Catégories de Réseaux sans fil.	18
Figure III-4 Le modèle des réseaux mobiles avec infrastructure.	19
Figure III-5 Exemple de réseau Ad-Hoc.....	20
Figure III-6 Anatomie générale d'un nœud de capteur.	23
Figure III-7 Exemples de nœuds de capteurs.....	23
Figure III-8 Architecture d'un réseau de capteurs.	25
Figure III-9 Les différents types de puits.....	25
Figure III-10 La pile protocolaire du réseau de capteur.	26
Figure III-11 Topologie hybride d'un réseau de capteurs sans fil.	28
Figure III-12 relation entre RCSF et système sensible au contexte.....	34
Figure IV-1 l'architecture des couches SAIL	38
Figure IV-2 Architecture des systèmes sensible au contexte.....	38
Figure IV-3 Framework du moteur contrôleur sensible au contexte.	39
Figure IV-4 Architecture de CONSORTS-S Platform.	40
Figure IV-5 Les services mobile Healthcare	40
Figure IV-6 Représentation d'un système avec un nœud portable.	41
Figure IV-7 la structure du système de la serre omniprésente.....	42
Figure IV-8 Architecture de BuL (niveau de construction) et les niveaux de WSN.	42
Figure V-1 relation entre faute, erreur et faille	45

Figure V-2 Classification des pannes	45
Figure V-3 Procédure de tolérance aux pannes	47
Figure V-4 Duplication avec comparaison	48
Figure V-5 TMR (triple modular redundancy).	49
Figure V-6 Système de contrôle industriel.	50
Figure V-7 les capacités d'un produit actif.....	53
Figure V-8 modèle fonctionnel de produit actif	54
Figure V-9 comportement autonome du produit actif	55
Figure V-10 Structure du réseau	64
Figure V-11 Forme d'un produit actif	64
Figure V-12 Scénario d'enregistrement d'un produit actif dans la station de base.	68
Figure V-13 Pseudo code de configuration des NRs.	69
Figure V-14 pseudo code de détermination des nœuds de chaque zone.....	70
Figure V-15 pseudo code de détermination des maîtres des zones	71
Figure V-16 diagramme de surveillance de compatibilité.....	72
Figure V-17 diagramme de surveillance de la distance.	73
Figure V-18 diagramme de surveillance interne.....	73
Figure V-19 pseudo code de la procédure de division des zones.	74
Figure V-20 division successive de zone.....	75
Figure V-21 pseudo code de détection du nœud présentant la faute de communication..	77
Figure V-22 résumé de notre contribution.....	78
Figure VI-1 Structure d'un nœud sous Castalia.....	81
Figure VI-2 Disposition des produits actifs dans un entrepôt 100m x 100m	83
Figure VI-3 Salutation entre les produits actifs	84
Figure VI-4 Cas des produits compatibles.....	84

Figure VI-5 Cas des produits incompatibles.....	85
Figure VI-6 cas du déclenchement d’alerte	85
Figure VI-7 cas du déclenchement de fausse alerte.....	86
Figure VI-8 détection du nœud défectueux	86
Figure VI-9 confirmation de la défaillance.....	87
Figure VI-10 Désactivation du nœud défectueux	87

LISTE DES ACRONYMES

BLR	Boucle Locale Radio.
CAS²	Context Awareness Service System.
CASN	Context aware sensornet.
CDMA	Code division multiple access.
EDGE	Enhanced Data Rates for GSM Evolution .
ESN	Enhanced Sensor network.
GPRS	General Packet Radio Service.
GPS	Global Positioning System.
GSM	Global System for Mobile Communications .
HiperLAN	<i>High Performance radio LAN.</i>
IEEE	Institute of Electrical and Electronics Engineers.
LEACH	Low Energy Adaptive Clustering Hierarchy
MEMS	Micro-electro-mechanical systems.
NA	nœud actif
NR	nœud redondant.
OSGi	Open Service Gateway Initiative.
OWL2	web ontologie langage.
PA	produit actif.
PDA	Personal Digital Assistant.
RCSF	Réseau De Capteur Sans Fil.
RDF	Resources Description Framework.
SAIL	sensor Abstraction and intégration layer.
SB	station de base
SD	Secure Digital.
SMECN	Small Minimum Energy Communication Network
SPIN	Software Process Improvement Network
UDP	User Datagram Protocol .
UMTS	Universal Mobile Telecommunications System.
Wi-Fi	Wireless Fidelity.
WLAN	Wireless Local Area Networks.
WMAN	Wireless Metropolitan Area Networks
WPAN	Wireless Personal Area Networks.
WSN	Wireless Sensor Network.
WWAN	Wireless Wide Area Networks.
XML	Extensible Markup Language.

CHAPITRE I. INTRODUCTION GENERALE

L'informatique, aujourd'hui, s'oriente vers l'informatique pervasive ou omniprésente, cette nouvelle forme de l'informatique rend l'information disponible partout et à tout moment. Ceci se fait d'une manière transparente pour l'utilisateur sans son intervention explicite et lui offre la possibilité de se concentrer sur sa tâche principale au lieu de configurer et de gérer l'ensemble des équipements informatiques mis à sa disposition, contrairement à l'informatique traditionnelle qui suppose qu'un utilisateur effectue une tâche définie dans un environnement déterminé.

L'informatique pervasive repose sur la connaissance du contexte de l'utilisateur pour lui délivrer le service approprié au moment opportun. La clé de cette sophistication est une prise de conscience du contexte d'utilisation et la capacité de communiquer entre les dispositifs, notons que le contexte est toute information qui décrit, à un moment donné, l'environnement dans lequel l'action aura lieu.

Pour mieux aider l'utilisateur dans ses tâches quotidiennes, les systèmes informatiques pervasifs doivent tenir compte du contexte global et adapter leurs services aux utilisateurs selon ce dernier. Cette aptitude est connue sous le nom de sensibilité au contexte.

La notion de sensibilité au contexte concerne l'utilisation du contexte dans les applications. Cette notion est une traduction de l'expression anglaise "context-awareness". Elle caractérise la capacité d'un système à s'adapter aux changements du contexte.

Nous parlons aussi du système sensible au contexte qui peut changer automatiquement ses services ou en déclencher comme réponse au changement de la valeur d'une information ou d'un ensemble d'informations qui caractérisent un service. Il peut découvrir et utiliser des informations contextuelles telles que la localisation de l'utilisateur, les caractéristiques de ses dispositifs, etc. Ce type de système se caractérise par une couche de captage du contexte qui est une collection de capteurs.

Un capteur est une source matérielle ou logicielle qui peut générer une information contextuelle. La performance des capteurs augmente lorsqu'ils collaborent pour accomplir les tâches du système, puisque lorsque les capteurs fonctionnent indépendamment, il arrive qu'un capteur tombe en panne, ce qui cause des dommages au système. D'où provient la notion d'intégration des réseaux de capteurs dans les systèmes sensibles au contexte.

Un réseau de capteurs sans-fil est considéré comme un type spécial des réseaux *Ad-Hoc* de façon que l'infrastructure fixe de communication et l'administration centralisée sont absentes et les nœuds jouent, à la fois, le rôle des hôtes et des routeurs. Les nœuds du réseau sont des capteurs intelligents, capables d'accomplir trois tâches complémentaires : la relève d'une grandeur physique, le traitement éventuel de cette information et la communication avec d'autres capteurs. Toute intervention humaine après le déploiement des capteurs est la plupart du temps exclue, le réseau doit donc s'autogérer.

Les réseaux de capteurs sont liés aux systèmes sensibles au contexte, au moins, de deux façons. D'une part, ces systèmes utilisent des capteurs pour acquérir des

informations contextuelles. D'autre part, les capteurs eux-mêmes peuvent être amenés à prendre conscience de leur propre contexte (la notion de capteurs sensibles au contexte dont le comportement peut être modifié dans la réponse au contexte actuel).

Problématique

Un système sensible au contexte se base sur les capteurs. Pour assurer la bonne performance de son fonctionnement, la solution opportune est l'intégration du RCSF dans ce système, et ceci grâce aux caractéristiques d'un RCSF. Donc, la question posée est : quelle est la contribution des RCSF dans un système sensible au contexte ?

Pour remédier à ce problème nous nous intéressons à la tolérance aux pannes dans les RCSF qui permet de gérer un réseau de façon transparente de façon que l'arrivée d'une panne n'empêche pas la continuité du fonctionnement du réseau de manière correcte.

Dans ce contexte, nous basons notre projet sur une application sensible au contexte à base d'un RCSF qui contrôle des produits dans un entrepôt puisque chaque produit est un produit actif doté des capacités d'interagir. La gestion de ces produits consiste à collecter des informations dans l'environnement des produits pour opérer des actions selon ces informations.

Nous proposons d'appliquer une méthode de correction de fautes au RCSF de cette application pour gérer les pannes survenant aux capteurs et minimiser l'intervention humaine. Cette méthode se base sur la redondance des nœuds dans le RCSF, lorsque les nœuds redondants sont en état passif.

Ainsi notre contribution est « la notion du produit actif tolérant aux pannes » : pour chaque produit, on lui associe un ensemble de nœuds de capteurs (un en état actif et les autres en état passif), et en cas de panne du nœud de capteur actif on le remplace par un autre du même ensemble associé au produit.

L'idée de la redondance, c'est pour renforcer le RCSF et faire augmenter la durée de vie de l'application. Le schéma suivant illustre le résumé des idées à réaliser :

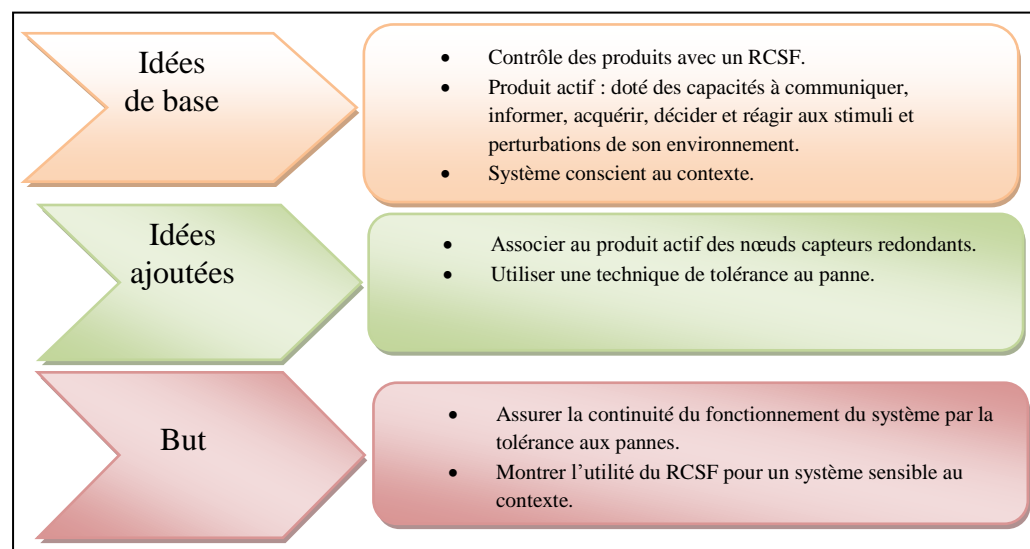


Figure I-1 Résumé Des Idées à Réaliser.

Le reste du mémoire s'organise comme suit :

Chapitre II : celui ci concerne les systèmes sensibles au contexte, il présente ses notions de base, leur architecture et leur conception.

Chapitre III : donne une présentation des réseaux de capteurs sans fil : leur type et caractéristiques, et autre notion importantes notamment leur relation avec les systèmes sensibles au contexte.

Chapitre IV : se rapporte à la classification des applications sensibles au contexte à base des réseaux de capteurs puis présente la description des exemples des travaux de chaque classe.

Chapitre V : ce chapitre se subdivise en trois parties : le premier concerne la tolérance aux fautes dans les RCSF, le deuxième concerne les travaux de bases pour notre contribution et le troisième explique notre contribution.

Chapitre VI : il concerne l'outil de simulation et l'implémentation de notre contribution.

Nous clôturons le mémoire par une conclusion générale qui résume notre travail.

CHAPITRE II. SYSTEMES SENSIBLES AU CONTEXTE

II.1. Introduction

Les systèmes embarqués ont connu une grande évolution technologique et les moyens de communication informatique ont incité les développeurs à intégrer les terminaux mobiles dans leurs applications, donnant ainsi naissance à de nouveaux systèmes d'information dits *pervasifs*. L'informatique pervasive est également connu par « informatique ubiquitaire » (En anglais *Ubiquitous Computing*). Ce terme a été proposé au départ par Mark Weiser, qui a également utilisé le mot « Omniprésente » ; Cette nouvelle tendance de l'informatique dont l'objet vise à assister implicitement et discrètement l'utilisateur dans les tâches qu'il accomplit au quotidien, devenant ainsi la base des systèmes informatiques diffus.

Dans ce chapitre, nous nous intéressons aux travaux existants pour les systèmes sensibles au contexte. Nous commençons par présenter les différentes définitions de la notion de contexte et les principes de la sensibilité au contexte (ou context-awareness), Nous présenterons ensuite l'architecture générale d'un système sensible au contexte et à la fin nous montrerons la relation entre les réseaux de capteur et les systèmes sensibles au contexte.

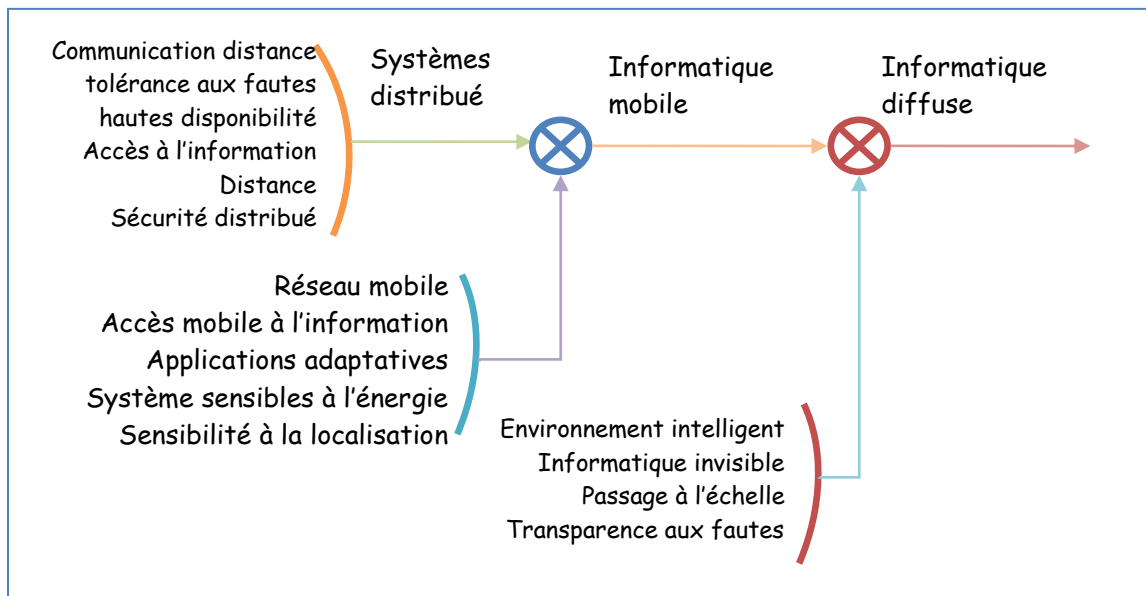


Figure II-1 Évolution vers l'informatique diffuse.

(MIRAOU, 2009)

II.2. Notion De Contexte

II.2.1. Notion De Contexte Dans Le Domaine De L'informatique Pervasif

Il existe une liste non exhaustive de définitions du contexte dans le domaine de l'informatique :

(Schilit et Theimer, 1994) étaient les premiers à proposer une définition du contexte comme étant la localisation de l'utilisateur, les identités et les états des personnes et objets qui l'entourent. Pour ces auteurs, le contexte, inclut la localisation et l'identité des personnes et des objets à proximité ainsi que les modifications pouvant intervenir sur ces objets. (Schilit, Adams et Want, 1994) Étudier le contexte, c'est répondre aux questions "On est où? ", "Avec qui ?", "Quelles sont les ressources qu'on utilise?"... Ils définissent donc le contexte comme les changements de l'environnement physique, de l'utilisateur et des ressources informatiques.

(Brown, Bovey et Chen, 1997) proposent la définition suivante : le contexte est l'identité de l'utilisateur, des personnes et les objets qui l'entourent, sa localisation géographique, son orientation, la saison, la température où il évolue... Ils restreignent le contexte aux éléments de l'environnement de l'utilisateur, puis ils introduisent l'heure, la saison, la température, l'identité et la localisation de l'utilisateur.

(Ryan, Pascoe et Morse, 1997) définissent le contexte en tant que localisation, environnement, identité, et temps de l'utilisateur.

(Dey, 2001) présente le contexte comme l'état émotionnel de l'utilisateur, son centre d'attention, sa localisation, son orientation, la date et le temps où il évolue, les objets et les gens qui existent dans son environnement. Puis *Dey* insiste sur la notion de pertinence de l'information en proposant une définition dans laquelle il essaie de préciser la nature des entités relatives au contexte.

(Ward et Hopper, 1997) voit le contexte comme étant les états possibles des environnements de l'application.

(Pascoe, 1998) définit le contexte comme un sous-ensemble d'états physiques et conceptuels ayant un intérêt pour une entité particulière.

(Winograd, 2001) propose la définition suivante : le contexte est un ensemble d'informations. Cet ensemble est structuré et partagé ; il évolue et sert l'interprétation. Il détaille cette définition en disant : "La considération d'une information comme contexte est due à la manière dont elle est utilisée et non à ses propriétés inhérentes".

(Henricksen, Indulska et Rakotonirainy, 2002) définissent le contexte comme étant la circonstance ou la situation dans laquelle une tâche informatique se déroule.

L'importance du contexte dans le domaine de l'interaction homme-machine et les systèmes mobiles a généré des définitions centrées sur l'utilisateur et d'autres sur l'application. Une analyse faite par (Brezillon, et al., 2004) concernant les définitions du terme contexte les a conduits à conclure que la plupart des définitions sont des réponses aux questions suivantes :

Qui ? Identité de l'utilisateur courant et d'autres personnes présentes dans l'environnement.

Quoi ? Percevoir et interpréter l'activité de l'utilisateur.

Où ? Localisation de l'utilisateur, ou d'un événement du système.

Quand ? Repère temporel d'une activité, indexation temporelle d'un événement, temps écoulé de la présence d'un sujet à un point donné.

Pourquoi ? Il s'agit de comprendre la raison d'être de l'activité.

Comment ? La manière de déroulement de l'activité.

II.2.2. Catégories de Contexte

Plusieurs chercheurs ont proposé des catégorisations du contexte selon différentes approches:

(Schilit, Adams et Want, 1994) et (Dey, 2001) ont catégorisé le contexte en deux classes:

contexte primaire il contient les informations sur la localisation, l'identité, le temps et l'activité (statut).

contexte secondaire il peut être déduit du contexte primaire (exemple : de la localisation, on peut déduire les personnes à proximité).

(Chen et Kotz, 2000) ont proposé deux catégories:

contexte actif qui influence le comportement d'une application.

contexte passif qui est nécessaire mais n'est pas critique pour l'application.

(Petrelli et al., 2000) a proposé la classification suivante:

contexte matériel comporte la localisation, la machine, la plateforme existante.

contexte social comporte les aspects sociaux comme la relation entre les individus.

(Hofer et al., 2003) a élaboré une catégorisation en deux classes :

contexte physique qui peut être mesuré par les capteurs physiques.

contexte logique qui contient les informations sur l'interaction (l'état émotionnel de l'utilisateur, ses buts, etc.)

Selon (CHAARI, 2007) les informations contextuelles utilisées dans la majorité des travaux existants peuvent être classées en quatre catégories principales :

- 1) **L'identité** : se réfère à la capacité d'assigner un identifiant à une entité qui doit être unique dans l'espace de nommage utilisé par les applications.
- 2) **La localisation** : elle peut concerner l'orientation, l'altitude et les relations spatiales entre les entités (comme les relations de proximité, de coexistence et de contenance).
- 3) **L'état ou activité** : encapsule les caractéristiques intrinsèques des entités qui interviennent dans le système. Par exemple, pour un lieu, l'état peut caractériser la température ambiante, la quantité de lumière existante ou le niveau de bruit courant. Pour une personne, l'état peut se référer à ses signes vitaux, sa fatigue ou son activité (par exemple, il est en train de conduire, lire, marcher, courir...). Pour les composants logiciels, l'état est tout attribut ou caractéristique qui peut être obtenue suite à une requête. Des exemples typiques de ces caractéristiques sont le temps de réponse d'un service, son taux d'utilisation, son état (disponible, activé, désactivé...).

- 4) **le temps** : peut aussi être une information contextuelle car il peut caractériser une entité. Le temps permet aussi d'établir un historique de valeurs permettant d'enrichir le contexte. En effet, l'enchaînement et l'ordonnement d'actions ou d'évènements dans le temps peuvent aussi être importants pour la décision prise par l'application.

II.2.3. Qualité du contexte

La qualité du contexte dépend de la source d'information (Ismail, 2010). Donc une application sensible au contexte doit exiger une certaine qualité des informations contextuelles. Dey propose quelques critères de qualité :

- ✓ L'exactitude (Accuracy).
- ✓ La tolérance du système vis-à-vis des fautes du senseur (Reliability).
- ✓ L'ensemble des valeurs possibles d'une information contextuelle (Coverage).
- ✓ Le changement exigé d'une information contextuelle (Resolution).
- ✓ Le temps nécessaire pour qu'une information contextuelle soit mise à jour (Frequency).
- ✓ Le temps entre la notification d'une information contextuelle et le changement de sa valeur (Timeliness).

Ebling et al. (Ismail, 2010) définit deux critères de qualité :

- ✓ La date de la dernière mise à jour (Freshness).
- ✓ Le taux d'exactitude (Confidence).

II.2.4. Utilisations du Contexte dans les Systèmes Informatiques

On peut citer les utilisations du contexte dans les systèmes informatiques suivantes : (MIRAOUI, 2009)

- 1) Senseur du contexte : où le contexte est capté et les informations décrivant le contexte courant (température, localisation, ...) sont présentés à l'utilisateur.
- 2) À associer le contexte aux données, encore appelé augmentation contextuelle. Par exemple les enregistrements sur les objets inspectés peuvent être associés à leur localisation ; les notes d'une réunion peuvent être associées aux personnes assistant à la réunion et le lieu où elle s'est déroulée.
- 3) Permettre la découverte de ressources contextuelles, par exemple, faire en sorte que l'impression d'un document ait lieu sur l'imprimante la plus proche.
- 4) Dans le cas des évènements survenant dans le contexte qui déclenchent les actions telles que le chargement de données cartographiques à l'entrée dans une région.
- 5) Médiation contextuelle : elle consiste à utiliser le contexte pour modifier un service. Par exemple pour décrire les limites et les préférences dans une large variété de données offertes, et ainsi pouvoir afficher les plus appropriées.
- 6) exécuter automatiquement une action selon le contexte dans lequel se trouve l'utilisateur (par exemple, transférer les appels d'un bureau à un autre selon les déplacements de l'utilisateur) (ABBAS, 2008).

II.2.5. Caractéristiques de L'information de Contexte

L'information du contexte a des caractéristiques différentes, nous citons les suivantes : (ELGHAYAM, 2011)

A. Le contexte est changeable avec le temps : l'information de contexte change continuellement de valeur avec le temps. Par exemple, le déplacement de l'utilisateur implique que le contexte « localisation de l'utilisateur » change de valeur.

B. L'information de contexte est hétérogène : L'hétérogénéité provient du fait que le contexte est capturé à partir d'une variété de sources. Particulièrement ce contexte peut être capturé (*Sensed*) directement par des capteurs physiques ou récupérés à partir de composants logiciels, donné par l'utilisateur (*Profiled*) ou dérivé (*Derived*) en synthétisant plusieurs sources ou interprété d'une seule source pour obtenir un niveau d'abstraction élevé de cette information.

C. L'information de contexte est imparfaite : cette caractéristique de contexte est déterminée selon la source de cette information. Le contexte est ambigu si des ressources séparées fournissent une même information avec des niveaux de granularités différents, le contexte peut être imprécis ou erroné ou même inconnu.

D. L'information de contexte est interdépendante : l'information de contexte peut être dépendante d'une autre information de contexte. Le changement de valeur d'une information de contexte peut influencer sur une autre valeur de contexte.

II.2.6. Délivrance du Contexte à l'Application

Il existe deux méthodes permettant de délivrer les informations de contexte à l'application. La première est basée sur une approche où l'application est conduite par les capteurs et le deuxième est basée sur la séparation entre l'acquisition du contexte et l'application (Ali, 2007) :

1) Approche conduite par les capteurs : Dans ce cas l'application est liée directement aux pilotes des capteurs. Elle doit alors implémenter le code se rapportant aux capteurs en utilisant les protocoles dictés par ces derniers. Cette approche a l'inconvénient de mélanger les détails de bas niveau de l'acquisition du contexte avec la sémantique de l'application. De plus, elle rend l'application très dépendante des capteurs et incapable d'interagir avec d'autres capteurs pour une même information de contexte.

2) Approche de séparation entre le contexte et l'application : Dans cette approche, l'acquisition du contexte est déléguée à un intergiciel qui se charge d'acquérir et de stocker les informations de contexte indépendamment de toutes les applications. Cette approche permet de cacher à l'application les détails se rapportant aux capteurs. L'application interagit avec cet intergiciel selon deux modes :

A. Mode par interrogation : qui consiste à envoyer une requête vers l'intergiciel pour récupérer la valeur du contexte. Ce mode est utile dans le cas où l'application utilise l'information de contexte une seule fois et suppose que l'application soit proactive, c'est à dire qu'elle sait quand demander l'information du contexte.

B. Mode par notification : qui consiste à s'inscrire auprès de l'intergiciel pour être notifiée par ce dernier de la valeur du contexte. Ce mécanisme est approprié pour un besoin répétitif du contexte. Dans ce cas, l'application peut établir des conditions qui précisent quand elle doit être notifiée.

Cette approche présente l'avantage de la réutilisation des mécanismes d'acquisition par plusieurs applications, ainsi que l'utilisation de plusieurs types de capteurs par une même application.

II.2.7. Sources d'informations

Selon (MIRAOU, 2009), il y a trois méthodes pour l'acquisition d'information pour les systèmes. Ces trois méthodes sont :

1) Capteurs physiques qui peuvent être intégrés ou non dans d'autres outils (cellulaire, PDA, ...) et permettant de capturer plusieurs types d'information physique telles que la température, la localisation géographique, niveau de bruit, lumière, etc.

2) Capteurs virtuels qui permettent d'extraire les informations contextuelles à partir des espaces virtuels tels que les programmes, les systèmes d'exploitation, le réseau, etc. (exemple : l'information sur la localisation peut être extraite à partir d'un calendrier électronique et d'un journal de connexion au réseau).

3) Capteurs logiques qui utilisent les informations des capteurs physiques et virtuels pour déduire d'autres informations.

De ce que précède (MIRAOU, 2009) a classé l'acquisition du contexte en trois méthodes qui sont :

Contexte capturé par les capteurs physiques (température, pression, lumière,..)

Contexte dérivé : acquisition ad hoc (temps, date).

Contexte fourni explicitement tel que le profil utilisateur, informations sur une machine dans un réseau, etc.

II.3. Notion de Sensibilité au Contexte

La caractéristique principale d'un environnement informatique diffus est le changement dynamique de leurs environnements ou bien plus précisément leurs contextes. Pour mieux aider l'utilisateur dans ses tâches quotidiennes, les systèmes informatiques diffus doivent tenir compte du contexte global et adapter leurs services aux utilisateurs selon ce dernier. Cette aptitude est connue sous le nom de sensibilité au contexte.

La notion de sensibilité au contexte concerne l'utilisation du contexte dans les applications. Cette notion est une traduction de l'expression anglaise "context-awareness". Elle caractérise la capacité d'un système à s'adapter aux changements du contexte.

Sensible au contexte est devenu un peu synonyme d'autres termes: adaptatif, réactif, sensible, situé, sensible au contexte, et de l'environnement dirigée (Krumm, 2009).

Le terme sensibilité au contexte a été évoqué pour la première fois par (Schilit, Adams et Want, 1994) dans leurs travaux sur un système de localisation. Ils ont défini la sensibilité au contexte comme l'aptitude d'une application à s'adapter au contexte de son exécution selon la localisation, l'ensemble des personnes à proximité, les machines, les équipements accessibles, de même que les changements de ces objets dans le temps.

(Brown, 1995) a défini la sensibilité au contexte dans son travail relatif à un guide touristique comme toute application qui prend en compte le contexte de l'utilisateur.

(Chen et Kotz, 2000) ont donné deux définitions :

a) Sensibilité active au contexte : une application qui s'adapte automatiquement au contexte découvert par le changement du comportement de l'application.

b) Sensibilité passive au contexte : une application qui présente le nouveau contexte ou celui mis-à-jour à un utilisateur intéressé ou rendre le contexte persistant pour une utilisation ultérieure.

«Un système est dit sensible au contexte s'il peut changer automatiquement ces formes des services ou déclencher un service comme réponse au changement de la valeur d'une information ou d'un ensemble d'informations qui caractérisent le service. »(MIRAOU, 2009).

les systèmes sensibles au contexte sont des systèmes qui peuvent adapter la présentation des informations afin de les rendre plus pertinentes au contexte d'un utilisateur en tenant compte, par exemple, des intérêts signalés par l'utilisateur ou de son contexte environnemental (SOUKKARIEH, 2010).

Les systèmes « sensibles au contexte » ou « adaptatifs au contexte » sont des systèmes qui peuvent découvrir et utiliser des informations contextuelles telles que la localisation de l'utilisateur, les caractéristiques de ses dispositifs, etc (SOUKKARIEH, 2010).

Les systèmes sensibles au contexte sont des systèmes capables de retourner à l'utilisateur des informations pertinentes et adaptées à ses besoins et son contexte qui influence son comportement lors de son interaction avec les systèmes d'information (SOUKKARIEH, 2010).

La sensibilité au contexte est la réaction du système suite à un contexte survenu. Cette réaction se fait en réalisant une compatibilité entre le sens de l'action et le sens particulier apporté par le contexte. Le sens de l'action peut être défini par l'objet de l'action, son objectif et les moyens disponibles pour invoquer cette action. La compatibilité est faite en contraignant le processus de construction et/ou d'exécution de ces actions. Un système sensible au contexte est un ensemble de mécanismes destinés pour, d'une part, la collection et la gestion des informations de contexte et, d'une autre part, la gestion et le contrôle des actions du système (son comportement) en fonction de ces informations de contexte (ELGHAYAM, 2011).

Un système est sensible au contexte s'il utilise le contexte pour améliorer son comportement pour devenir plus approprié au contexte de l'utilisateur. Le comportement inclut les comportements visibles, y compris ceux qui ne sont accessibles qu'après le fait, comme le rappel contextuel en utilisant l'augmentation des données stockées ; et les comportements des systèmes qui ne sont visibles que par les effets secondaires, une plus grande vitesse ou la durée de vie de la batterie (Chalmers, 2011).

II.3.1. Classes de Sensibilité au Contexte

Selon Dey et Abowd (CHAARI, 2007), un système est sensible au contexte s'il utilise le contexte pour fournir des informations et des services pertinents pour l'utilisateur, où la pertinence dépend de la tâche demandée par l'utilisateur. Cette définition d'un système sensible au contexte a été adoptée par les chercheurs dans ce domaine. Elle a mis en évidence trois catégories:

1) Présentation d'information et services : se rapporte aux applications qui présentent des informations du contexte. Ces applications peuvent proposer des choix d'actions appropriées à l'utilisateur. Il y a plusieurs exemples de ce type de travaux dans la littérature et dans quelques systèmes commercialisés. Par exemple, il y a des applications pour montrer la localisation de l'utilisateur ou de son véhicule sur une carte. Elles peuvent aussi proposer des icônes (ou des alertes) des centres d'intérêts voisins de l'utilisateur.

2) Exécution automatique de services : décrit les applications qui déclenchent une commande, ou reconfigurent le système à la place de l'utilisateur selon les changements de contexte. Dans cette catégorie on cite : le système Téléport (CHAARI, 2007) qui assure le transport automatique de profil utilisateur lorsqu'il passe d'une machine à une autre, un système d'enregistrement automatique de son quand une réunion ou un rassemblement non planifié se passe dans un certain lieu, des téléphones mobiles qui changent leurs comportement et leurs configurations (vibreur/sonnerie) selon l'environnement de l'utilisateur, un système de sécurité portable qui détecte si l'utilisateur est effrayé en utilisant des capteurs biométriques, et des dispositifs qui fournissent des signaux de rappel quand les utilisateurs sont à un lieu précis.

3) Stockage d'information selon le contexte : les applications associent des données au contexte de leur utilisation. Par exemple, dans une conférence, une application étiquette des notes prises par l'utilisateur avec le lieu et le temps de l'observation. Dans des domaines similaires, Time-Machine Computing et Placeless Documents (CHAARI, 2007) sont deux systèmes qui attachent l'identité des utilisateurs, leurs lieux et le temps de création et d'utilisation de ressources logicielles afin de pouvoir les utiliser d'une façon plus facile et rapide ultérieurement. D'autres exemples plus complexes dans cette catégorie sont des applications d'aide-mémoire telles que Forget-Me- Not et Rememberance Agent (CHAARI, 2007).

Chalmers, dans (Chalmers, 2011), a identifié sept utilisations de l'information contextuelle:

- 1) **Affichage de contexte :** lorsque le contexte est détecté il est présenté à l'utilisateur, par exemple, affichage d'emplacement actuel.
- 2) **Augmentation contextuelle :** Elle annote les données avec le contexte de sa génération. Par exemple, des notes de réunion peuvent être associées avec les participants et le lieu de la réunion. Les données doivent être stockées de façon que toute l'indexation ou la recherche future soit simplifiée.
- 3) **Configuration consciente au Contexte :** par exemple, pour provoquer l'impression pour qu'elle se fasse sur l'imprimante la plus proche, ou provoquer la sélection de procurations à proximité en cas de besoin. Le modèle doit aider à répondre à des questions telles que "trouver le plus proche", par exemple, le cas du déplacement d'un écran à un écran près de l'utilisateur.
- 4) **Contexte déclenchant des actions :** c'est le cas, par exemple, du chargement automatique des données de la carte pour l'emplacement à côté prédit (bien que cela puisse également être considérée comme entrée plutôt que contexte); ou pause de la télévision lorsque le téléphone sonne.
- 5) **Médiation contextuelle :** c'est l'utilisation du contexte pour modifier les services fournis ou les données demandées pour répondre au mieux aux besoins et limites découlant du contexte de l'interaction.

- 6) **Adaptation contextuelle** : l'adaptation de l'environnement peut être vue comme une forme de contexte déclenchant des actions, mais dans cette classe de la conscience de contexte, il considère une application dont les comportements se manifestent dans l'environnement (éclairage, chauffage, systèmes de médias, etc.) plutôt que d'une application "calcul" qui fait l'objet d'une tâche.
- 7) **Présentation consciente au contexte** : se réfère à l'adaptation de l'interface utilisateur ou de la présentation des données, par exemple, le réglage de widgets d'interaction selon l'affichage des dispositifs, ou le choix des notifications de message en fonction du contexte. Cette utilisation de contexte est étroitement alignée sur la gestion des préférences de l'utilisateur.

II.3.2. Architecture d'un Système Sensible au Contexte

Dans le domaine de la sensibilité au contexte, les chercheurs ont proposé des architectures différentes pour un système sensible au contexte, ces architectures se diffèrent dans les fonctions, les noms et l'emplacement de leurs couches. Mais toutes ces architectures se basent sur cinq couches principales (CHAARI, 2007)



Figure II-2 Architecture générale d'un système sensible au contexte.

1) Couche de capture du contexte : elle est composée d'une collection de capteurs. Un capteur est une source matérielle ou logicielle qui peut générer une information contextuelle. Il y a trois types de capteurs : physiques, virtuels et logiques.

A. Capteurs physiques : sont des dispositifs matériels qui sont capables de fournir des données de contexte.

B. Capteurs virtuels : ils fournissent des informations contextuelles à partir d'applications ou services logiciels.

C. Capteurs logiques : Ce type de capteurs utilise généralement plusieurs sources d'information contextuelles pour fournir une autre information de synthèse plus précise. Ces capteurs peuvent réutiliser des capteurs physiques et virtuels pour fournir un contexte de plus haut niveau. Dans certains travaux, les capteurs logiques sont considérés comme des interpréteurs de contexte.

2) Couche d'Interprétation et d'agrégation du contexte : Cette couche offre des moyens d'interprétation des données contextuelles fournies par les capteurs du contexte. Elle sert à l'analyse et à la transformation des données brutes fournies par la couche de capture du contexte dans d'autres formats de haut niveau qui sont plus faciles à manipuler et à utiliser. La complexité des interprétations de contexte peut varier d'une simple agrégation de valeurs qui proviennent de plusieurs capteurs à des raisonnements

ou analyses statistiques complexes. Cette couche doit aussi assurer la résolution de conflits causés par l'utilisation de plusieurs sources de contexte.

3) Couche de stockage et historique du contexte : organise les données capturées et interprétées et les stocke pour une utilisation ultérieure. Ce stockage peut être centralisé ou distribué.

A. Solution centralisée : est l'option la plus répandue et la plus utilisée puisqu'elle facilite la gestion des mises à jour et des variations des valeurs du contexte.

B. Gestion distribuée du contexte est beaucoup plus complexe puisqu'elle inflige des fonctions additionnelles de découvertes de ressources et d'actualisation des valeurs du contexte. De plus, cette gestion distribuée alourdit la tâche de l'application qui doit gérer la collecte des différentes informations contextuelles d'une façon interne.

4) Couche dissémination du contexte : Cette couche assure la transmission des différentes informations contextuelles à l'application. Elle assure une transparence totale de la communication avec l'application. En conséquence, le développement de l'application devient plus simple. La couche de dissémination du contexte offre des moyens de communication standards pour notifier l'application des changements de contextes et leur transmission à l'application.

Plusieurs systèmes offrent des mécanismes de gestion d'événements qui se basent essentiellement sur les fonctions de gestion de requêtes directes ou de notification. L'application peut demander un accès direct à une information contextuelle précise (pull) mais elle peut aussi s'abonner pour recevoir tous les changements des valeurs de cette information (publish/subscribe). Ces deux fonctions principales assurent un moyen de communication transparent et efficace pour la dissémination des valeurs de contexte à l'application.

5) Couche application : dans les systèmes sensibles au contexte existants est représentée par l'application qui offre ses services aux différents clients concernés. Cette couche est responsable de l'extraction des informations des différents sources de données attachées à l'application. Elle doit aussi implémenter les réactions nécessaires aux changements du contexte. Chaque application s'abonne à la couche de dissémination du contexte pour accéder aux différentes informations contextuelles et être informée de leurs changements.

Une application peut accéder à ces informations de deux façons différentes : synchrone et asynchrone. Dans le cas d'un accès synchrone, l'application demande à la couche de dissémination de lui fournir une information contextuelle précise. Ceci est réalisé généralement par des appels directs de fonctions au niveau de la couche dissémination. Dans le cas de la communication asynchrone, l'application s'abonne à des événements spécifiques qui correspondent à des changements de valeurs de contexte.

II.3.3. Modèles de Contexte

Les structures de données utilisées pour encoder et transmettre le contexte appartiennent le plus souvent à l'une des six catégories suivantes (POPOVICI, 2012):

1. **Modèle clef-valeur :** Une clef représente un attribut de l'environnement, dont il faut mesurer la valeur. Par exemple, <Température_bureau_4, 20> représente l'information de température avec le modèle clef-valeur. Ce modèle est simple et facile à entretenir, mais peut s'avérer peu flexible.

2. **Modèle à balises** : Le modèle à balises contient des données organisées dans des structures hiérarchiques, grâce aux balises. L'une des plus importantes utilisations de ce modèle dans l'informatique mobile sont les profils (utilisateur, terminal).
3. **Modèle orienté objet**. Dans ce modèle, les informations de contexte sont capturées dans les états des objets. De plus, les méthodes des objets permettent d'accéder aux informations et de faire des traitements spécifiques.
4. **Modèle basé sur la logique**. Dans ce modèle, les données de contexte peuvent être exprimées comme des faits dans un système à base de règles. Des méthodes de la logique de premier ordre peuvent être appliquées pour raisonner sur le contexte.
5. **Ontologie** : Des ontologies et d'autres outils du *Semantic Web* (comme le langage OWL2) peuvent être utilisés pour représenter et traiter le contexte. Les ontologies permettent de créer des vocabulaires partagés entre les systèmes et d'ajouter de la sémantique aux données de contexte.
6. **Logique de situation** : Modéliser les informations de contexte en fonction des situations.

Les applications s'exécutant sur un ordinateur de bureau ou un serveur peuvent utiliser des modèles basés sur la logique ou sur des ontologies afin de déduire de nouvelles informations. Les applications pour les téléphones mobiles nécessitent des modèles plus légers, leur permettant de réagir vite aux changements de contexte.

II.3.4. Conception des Applications Sensibles au Contexte

La conception d'une application sensible au contexte peut être divisée en cinq étapes, dans chacune d'elle il y a un ensemble de tâches à réaliser : (NAILI, 2012)

1) Spécification : Dans cette étape, le concepteur doit définir les domaines d'utilisation de cette application, l'ensemble des utilisateurs ciblés et les services à fournir, afin de spécifier les types de contexte et le comportement selon chaque cas.

2) Acquisition : Cette étape est celle où le concepteur doit déterminer le matériel et les logiciels des capteurs nécessaires pour acquérir, sauvegarder et combiner si possible le/les contexte(s) déterminés dans la première étape ; ceci inclut la plateforme sur laquelle le logiciel va être installé, les APIs nécessaires pour établir la communication avec les capteurs afin d'utiliser les informations fournies par eux dans le processus.

3) Livraison : Cette étape concerne la façon avec laquelle le contexte doit être livré par les capteurs à l'application.

4) Réception : L'application doit spécifier le/les contextes intéressant(s) et par quel capteur. Ceci se fait par l'interprétation des informations fournies par les capteurs afin de convertir ces informations en une forme utile, ensuite l'analyse du contexte afin de voir si la combinaison de ce contexte avec d'autres va décrire la situation réelle de l'utilisateur.

5) Action : Dans cette dernière étape, tous les contextes reçus doivent être analysés afin de déterminer le comportement (qui est sensible au contexte) adéquat et l'exécute.

II.4. Conclusion

Dans ce chapitre nous avons abordé la notion du contexte d'un système et les systèmes sensibles au contexte. Nous avons présenté les catégories du contexte, ses utilisations dans un système informatique, les caractéristiques de l'information de contexte, la sensibilité au contexte et ses classes, ensuite nous avons présenté l'architecture d'un système sensible au contexte, sa conception et les modèles du contexte.

Le système sensible au contexte collecte les informations de son contexte en utilisant les capteurs. Pour un bon fonctionnement du système, les capteurs doivent travailler en collaboration d'où l'utilisation des RCSF est très importante dans le système sensible au contexte.

Le chapitre suivant est consacré à l'étude des RCSF.

CHAPITRE III. RESEAUX DE CAPTEURS SANS FIL

III.1. Introduction

Les avancées récentes dans le domaine de communication sans fil et les technologies « MEMS » permet de remplacer l'architecture de réseau de capteurs classique, qui repose sur des chemins de câbles parfois complexes. Par conséquent non rentable du point de vue encombrement et coût.

Les MEMS ont permis le développement des microcomposants qui intègrent des dispositifs de captages et de communication sans fil dans un seul circuit, à dimension réduite, et avec un coût raisonnable.

Ces composants, communément appelés micro-capteurs, ont favorisé l'idée de développer les Réseaux de Capteurs Sans Fils (*RCSF*) ou *WSN* (Wireless Sensor Network) basés sur l'effort collaboratif d'un grand nombre de nœuds opérant d'une façon autonome et communiquant entre eux via des transmissions à courte portée.

III.2. Réseaux Sans Fils

III.2.1. Définition

Un réseau sans fil (wireless network) est un réseau où au moins deux nœuds peuvent avoir une communication sans liaison filaire. Les réseaux sans fil sont ceux qui utilisent le canal air pour communiquer en utilisant les ondes hertziennes, les infrarouges ou le laser. Ces réseaux de communications permettent aux utilisateurs de profiter de tous les services traditionnels des réseaux indépendamment de leurs positions géographiques.

L'utilisation des réseaux sans fil procure plusieurs avantages, notamment (DOYI, 2010):

- ✓ L'usage facile dans les endroits à câblage difficile.
- ✓ La réduction du temps de déploiement et d'installation.
- ✓ La réduction des coûts d'entretien.
- ✓ L'augmentation de la connectivité.
- ✓ La réduction de l'encombrement.
- ✓ La portabilité, le nomadisme et même la mobilité.
- ✓ etc.

III.2.2. Classification des Réseaux Sans Fil

Les réseaux sans fil peuvent avoir une classification selon deux critères. Le premier est la zone de couverture du réseau. Au vu de ce critère il existe quatre catégories : les réseaux personnels, les réseaux locaux, les réseaux métropolitains et les réseaux étendus. Le second critère est celui de l'infrastructure ainsi que le modèle adopté. Par rapport à ce critère on peut diviser les réseaux sans fils en : réseaux avec infrastructures et réseaux sans infrastructure, comme on le voit dans l'illustration de la figure III-1. (Amina, 2010)

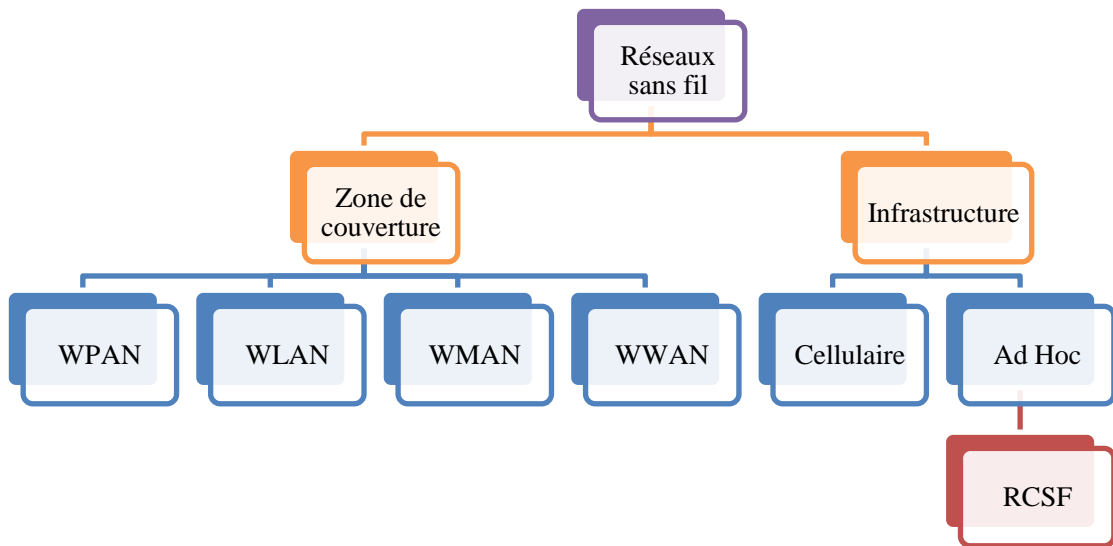


Figure III-1 Classification des réseaux sans fil

1) Classification selon la zone de couverture. La classification des réseaux en fonction de la taille de la zone qu'ils couvrent, donne quatre classes des réseaux qui sont les suivants : WPAN, WLAN, WMAN, WWAN :

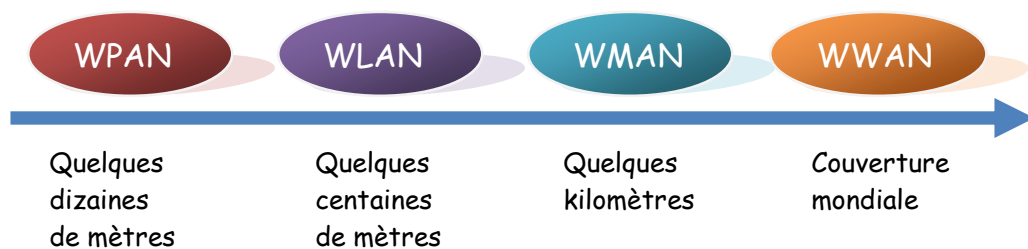


Figure III-2 Classification des réseaux sans fil suivant leur taille.

A. Les WPAN (Wireless Personal Area Networks). Dans cette catégorie, on retrouve les réseaux sans fil à l'échelle humaine dont la portée maximale est limitée à quelques dizaines de mètres autour de l'utilisateur (bureaux, salles de conférence...). Ils sont le plus souvent utilisés à faire communiquer entre eux des matériels présents sur une personne (par exemple une oreillette et un téléphone portable). Ils sont également utilisés

pour relier des équipements informatiques entre eux sans liaison filaire. On y trouve les standards tels que le Bluetooth, ZIGBEE, RFID et HomeRF.

B. Les WLAN (Wireless Local Area Networks). C'est la catégorie des réseaux locaux sans fil dont la portée va jusqu'à 500 m, pour les applications couvrant un campus, un bâtiment, un aéroport, un hôpital, etc. On y trouve les standards tels que le Wi-Fi (Wireless Fidelity) et les HIPERLAN.

C. Les WMAN (Wireless Metropolitan Area Networks). Plus connus sous le nom de Boucle Locale Radio (BLR), ce type de réseau utilise le même matériel que celui qui est nécessaire pour constituer un WLAN mais peut couvrir une plus grande zone de la taille d'une ville avec une portée pouvant aller jusqu'à 50 Km. C'est dans cette catégorie que l'on classe le WiMAX et les HIPERMAN.

D. Les WWAN (Wireless Wide Area Networks). C'est la catégorie de réseaux cellulaires mobiles dont la zone de couverture est très large, à l'échelle mondiale. Dans cette catégorie, on peut citer le GSM et ses évolutions (GPRS, EDGE), le CDMA et l'UMTS.

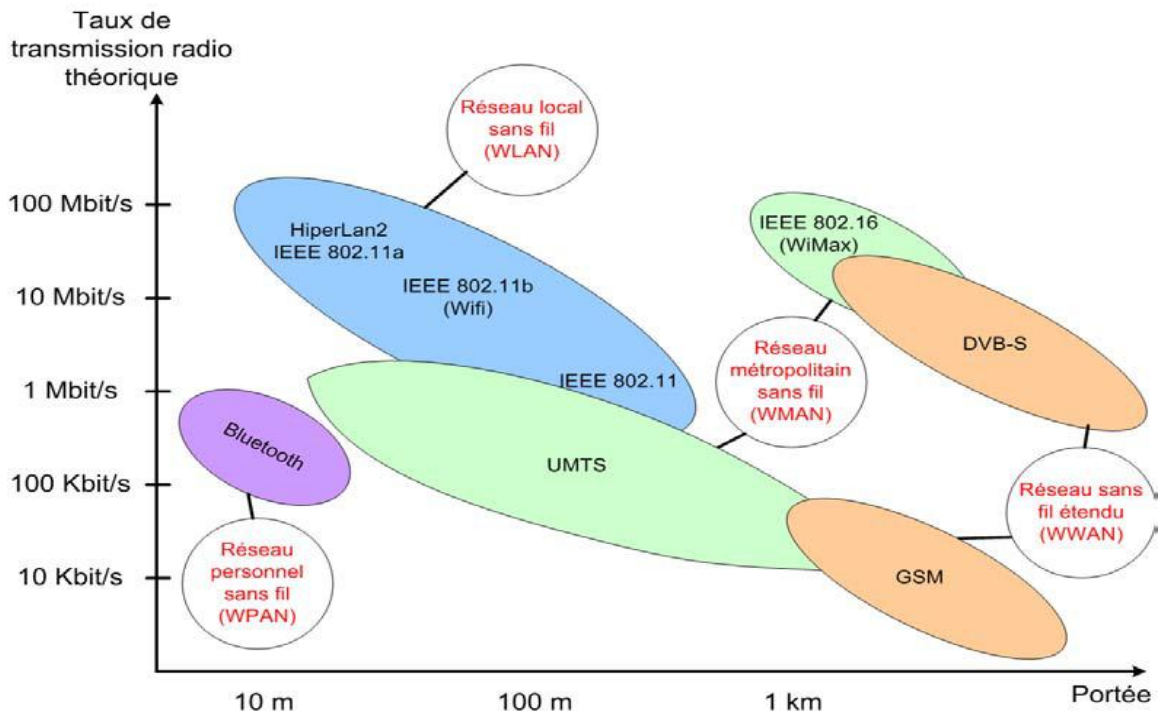


Figure III-3 Catégories de Réseaux sans fil.

2) Classification selon l'infrastructure. Les environnements mobiles sont des systèmes composés de sites mobiles et qui permettent à leurs utilisateurs d'accéder à l'information indépendamment de leurs positions géographiques. Les réseaux mobiles ou sans fil, peuvent être classés en deux classes: *les réseaux avec infrastructure* et *les réseaux sans infrastructure*.

A. Les réseaux avec infrastructure. Les réseaux sans fil avec infrastructure sont également appelés réseaux cellulaires. Ce type de réseaux se compose des éléments suivants:

- 1- Les "sites fixes" du réseau filaire.
- 2- Les "sites mobiles", réseaux sans fils.

Certains sites fixes, appelés stations de base (**SB**). Les points d'accès sont des terminaux fixes reliés en général à une source d'énergie illimitée et chargés de relayer les informations qui circulent dans leur cellule. Ils jouent le rôle de serveur pour chaque terminal mobile présent dans la cellule. Ils sont munis d'une interface de communication sans fil pour la communication directe avec les sites mobiles localisés dans une zone géographique limitée, appelée cellule comme le montre la figure suivante :

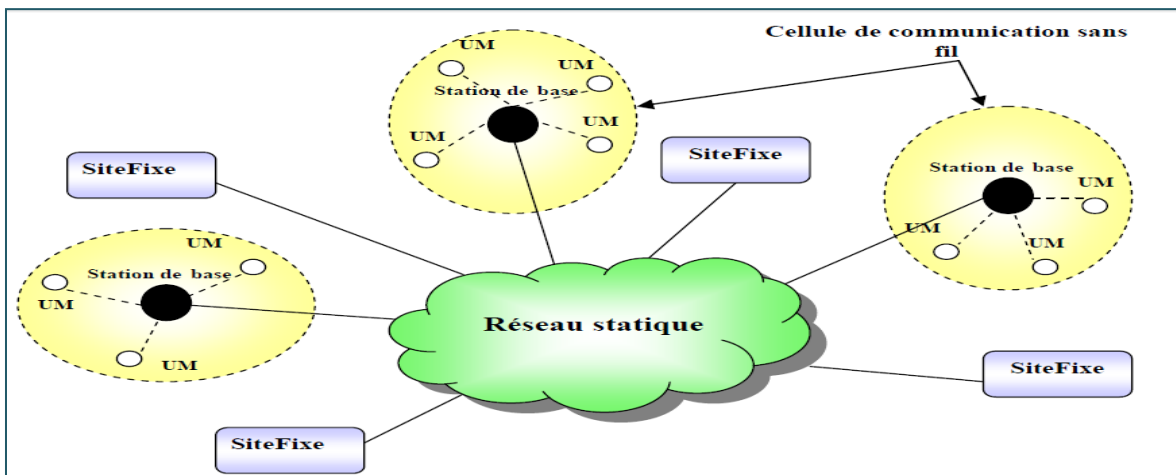


Figure III-4 Le modèle des réseaux mobiles avec infrastructure.

A chaque station de base correspond une cellule à partir de laquelle des unités mobiles peuvent émettre et recevoir des messages. Les sites fixes sont interconnectés entre eux à travers un réseau de communication filaire. Une unité mobile ne peut être, à un instant donné, directement connectée qu'à une seule station de base. Elle peut communiquer avec les autres sites à travers la station à laquelle elle est directement rattachée. Ces réseaux sont donc une réponse au problème d'affaiblissement du signal. En effet, en déployant plusieurs points d'accès on peut étendre la couverture globale du réseau. Tous les points d'accès sont reliés entre eux par un réseau filaire qui se charge de la partie routage ainsi que de la plupart des fonctions d'administration (authentification centralisée par exemple).

Dans cette catégorie, on trouve les réseaux WLAN (Wi-Fi), WMAN (WiMAX) et WWAN (GSM).

B. Les réseaux sans infrastructure (Ad hoc). En mode ad hoc, il n'y a pas de point d'accès fixe, l'infrastructure n'est composée que des stations elles-mêmes, ces dernières jouant à la fois le rôle de terminaux et de routeurs pour permettre le passage de l'information d'un terminal vers un autre sans que ces terminaux soient reliés directement.

La caractéristique essentielle d'un réseau ad-hoc est l'existence de tables de routage dynamiques dans chaque nœud. C'est la catégorie des réseaux WPAN tels que le Bluetooth.

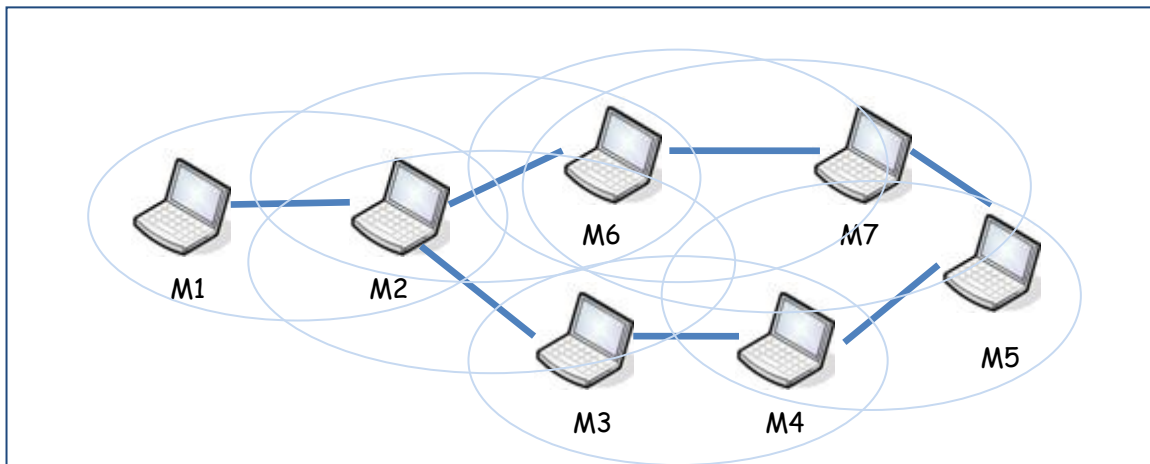


Figure III-5 Exemple de réseau Ad-Hoc.

On peut distinguer six grandes caractéristiques des réseaux Ad Hoc : (Amina, 2010)

1) La topologie est dynamique : C'est une conséquence directe de la mobilité des unités qui composent le réseau. Le tracé des routes peut changer pendant l'émission d'un paquet.

2) Les liens sont asymétriques : En effet, la liaison entre deux unités n'est pas forcément bidirectionnelle.

3) La bande passante est limitée : car les communications par voies hertziennes imposent un partage du médium de communication entre les différents hôtes. On appelle ce phénomène réutilisation spatiale. Ce phénomène est dû à l'atténuation des signaux avec la distance qui fait que même si un médium peut être utilisé simultanément par plusieurs hôtes, il peut y avoir des collisions et donc la nécessité de procéder à des réémissions.

4) Fortes contraintes énergétiques : Cela est dû au fait que chacune des unités doit bien souvent embarquer une alimentation autonome.

5) Absence d'infrastructure et de contrôle centralisé : En effet, les hôtes sont mobiles et c'est à eux qu'il incombe de maintenir les connexions.

6) Les interférences : Il y a de nombreuses interférences entre les hôtes du réseau ou encore d'une onde avec elle-même (cas d'une réflexion d'onde). Elles accroissent le nombre d'erreurs sur la transmission et imposent un amoindrissement des performances.

III.3. Réseaux de Capteurs Sans Fil

III.3.1. Capteurs Sans Fil

III.3.1.1. Définition d'un capteur sans fil :

Les capteurs sans fil sont des petits appareils dotés d'une batterie, capable de traiter des informations et de les transmettre, via les ondes radio et de détecter des événements s'ils se trouvent à l'intérieur de leur rayon de perception.

Les capteurs intelligents (Smart Sensors) sont des dispositifs matériels dans lesquels coexistent le(s) capteur(s) et les circuits de traitement et de communication. Leurs relations avec des couches de traitement supérieures vont bien au-delà d'une simple « transduction de signal ».

Les capteurs intelligents sont des « capteurs d'informations » et non pas simplement des capteurs et des circuits de traitement du signal juxtaposés. De plus, les « Smart Sensors » ne sont pas des dispositifs banalisés car chacun de leurs constituants a été conçu dans l'objectif d'une application bien spécifique.

La différence entre le capteur intelligent et le capteur classique réside dans la capacité de calcul interne et son interface de communication bidirectionnelle.

III.3.1.2. Caractéristiques des Capteurs Sans Fils :

Un capteur sans fil est doté des caractéristiques suivantes (lamine, 2007):

- Capable de calculer.
- Capable de communiquer.
- Capte toujours.
- Préposition / déploiement aléatoire.
- Limitation de la durée de vie des batteries.
- Densité (petit / grand nombre).
- La rapidité : c'est le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant où l'information prise en compte par la partie commande.
- L'étendue de la mesure : c'est la différence entre le plus petit signal détecté et le plus grand perceptible sans risque de destruction pour le capteur.
- La sensibilité : c'est la plus petite variation d'une grandeur physique que peut détecter un capteur.

En analysant la gamme des composants disponibles sur le marché et les prototypes présents dans la littérature, il est évident que la principale caractéristique d'un nœud de capteurs sans fil est sa *petite taille*. Une deuxième caractéristique, évidente mais essentielle, est l'*autonomie* (pas seulement du point de vue de leur source d'énergie, mais aussi de leur fonctionnement). Ces deux premières particularités induisent plusieurs autres caractéristiques à considérer, en particulier la vitesse de calcul et la vitesse de transmission. Des performances élevées en termes de vitesse de traitement et de transmission impliquent une consommation d'énergie élevée.

De manière générale, il est souhaitable que la durée de vie de la batterie d'un capteur soit la plus longue possible, donc les différentes unités qui composent un capteur sont généralement très limitées en termes de ressources et de performance pour que leur consommation d'énergie soit extrêmement faible.

D'autres caractéristiques sont souvent utilisées comme spécificités des nœuds de capteurs dans la bibliographie, par exemple qu'ils aient un faible coût de production (Duran-Faundez, 2009).

III.3.1.3. Architecture d'un Nœud de Capteur

Un nœud capteur est composé de plusieurs éléments ou modules correspondant chacun à une tâche particulière : captage, traitement, transmission de données, il comprend également une source d'énergie comme l'illustre la Figure 6: (BEYDOUN, 2009)

1) Unité de captage : composée d'un moyen de captage appelé « capteur » qui obtient des mesures sur les paramètres environnementaux et d'un convertisseur Analogique/Numérique qui convertit l'information relevée et la transmet à l'unité de traitement.

2) Unité de traitement : composée d'un processeur et d'une mémoire intégrant un système d'exploitation spécifique. Cette unité possède deux interfaces, une interface pour l'unité de captage et une interface pour l'unité de communication.

Elle acquiert les informations en provenance de l'unité de captage et les envoie à l'unité de communication. Cette unité est chargée aussi d'exécuter les protocoles de communications qui permettent de faire collaborer le capteur avec d'autres capteurs. Elle peut aussi analyser les données captées.

3) Unité de transmission: unité responsable de toutes les émissions et réceptions de données via un support de communication radio. Elle peut être de type optique, ou de type radiofréquence.

4) Unité d'énergie : un capteur est muni d'une batterie pour alimenter tous ses composants. Cependant, à cause de sa taille réduite, la batterie dont il dispose est limitée et généralement irremplaçable. Pour cela, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs.

5) Unités additionnelles : Il existe des capteurs qui sont dotés d'autres unités additionnelles comme le système de positionnement GPS (Global Positioning System) et un mobilisateur lui permettant le déplacement.

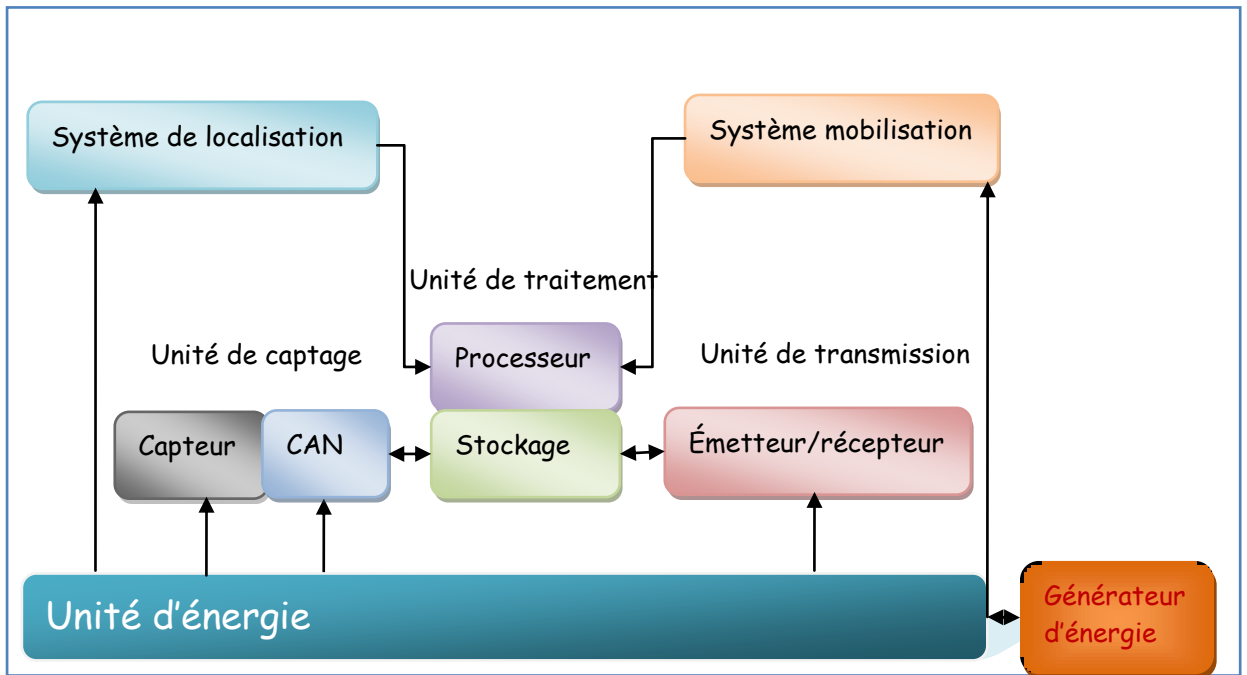


Figure III-6 Anatomie générale d'un nœud de capteur.

(Zouinkhi, 2011)

III.3.1.4. Images de quelques capteurs :

Il ya une multitude de modèles des capteurs en relation avec l'application a laquelle il est destiné. Parmi les modèles les plus courants, on trouve par exemple le capteur "weC" de l'université de Berkeley pour capter la température et la luminosité. La figure III-7 montre les images de quelques capteurs (lamine, 2007).

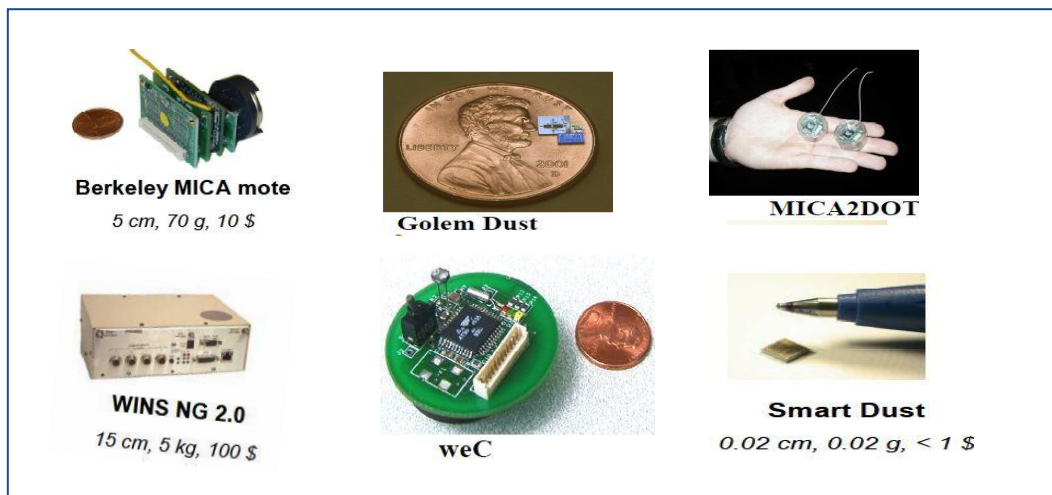


Figure III-7 Exemples de nœuds de capteurs

III.3.2. Description des réseaux de capteurs sans fil

III.3.2.1. Définition d'un réseau de capteurs sans fil

Les réseaux de capteurs sans fil sont considérés comme un type spécial des réseaux *Ad-Hoc* où l'infrastructure fixe de communication et l'administration centralisée sont absentes et les nœuds jouent, à la fois, le rôle des hôtes et des routeurs.

Les nœuds capteurs sont des capteurs intelligents "smart sensors", capables d'accomplir trois tâches complémentaires : la relève d'une grandeur physique, le traitement éventuel de cette information et la communication avec d'autres capteurs.

L'ensemble de ces capteurs, forme un *réseau de capteurs*. Les capteurs sont placés de manière plus ou moins aléatoire dans des environnements pouvant être dangereux. Toute intervention humaine après le déploiement des nœuds capteurs est la plupart du temps exclue, le réseau doit donc s'autogérer.

Afin que les nœuds capteurs travaillent d'une façon coopérative, les informations recueillies sont partagées entre eux par voie hertzienne. Le choix du lien radio plutôt que du lien filaire permet un déploiement facile et rapide dans un environnement pouvant être inaccessible pour l'être humain (Iamine, 2007).

Les réseaux de capteurs peuvent être composés, suivant leur utilisation, de différents types de nœuds capteurs, tels que les capteurs sismiques, thermiques, visuels, infrarouges, acoustiques et radar, ils sont capables de surveiller une grande variété de phénomènes ambiants, notamment: Température, Humidité, Mouvement des véhicules, la pression, les taux de bruits, la présence ou l'absence de certains types d'objets, le taux de frottement sur des objets attachés, et d'autres caractéristiques tel que la vitesse, la direction et le volume d'un objet donné. (Khelladi, 2004)

Les nœuds capteurs peuvent être utilisés pour la surveillance continue d'un phénomène, la détection et l'identification d'un événement, ou la surveillance d'un terrain et le contrôle local d'objets déclencheurs d'évènements (actionneurs).

III.3.2.2. Architecture globale d'un réseau de capteurs sans fil

Un réseau de capteurs sans fil générique est composé d'un grand nombre de nœuds capteurs dispersés dans le terrain d'intérêt appelé *champ de captage*.

Les **nœuds** ont la possibilité de collecter périodiquement les données sur le phénomène surveillé. Les nœuds envoient les rapports de captage à un nœud responsable et spécial appelé *puits* (sink). Le nœud puits est responsable, en plus de la collecte des rapports, de la diffusion des demandes sur les types de données requise aux capteurs via des messages de requêtes. Un réseau de capteurs peut contenir plusieurs nœuds puits diffusant des intérêts différents. un capteur doit pouvoir stocker toutes les requêtes reçues, et les traiter séparément (Khelladi, 2004).

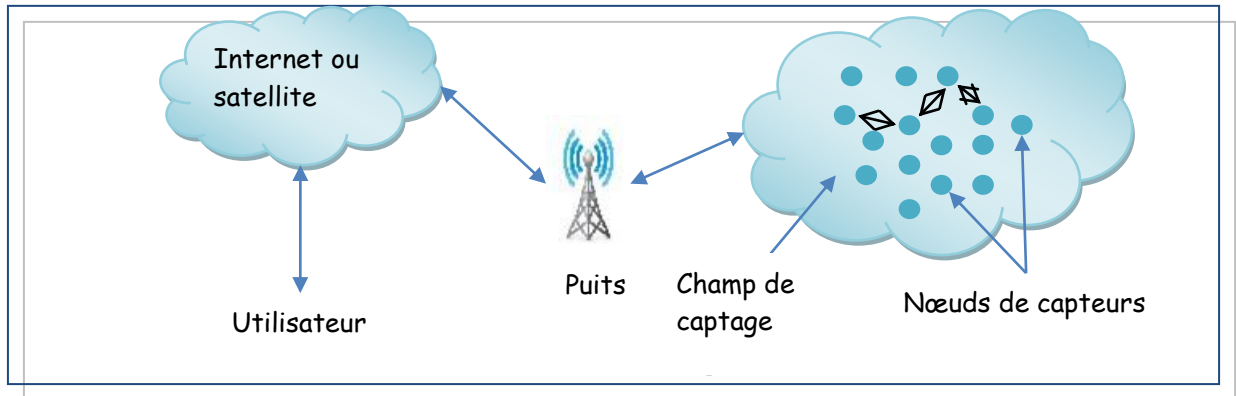


Figure III-8 Architecture d'un réseau de capteurs.

(Khelladi, 2004)

Il y a essentiellement trois types de puits (Sink) :

- 1) Un nœud appartenant au réseau comme n'importe quel autre nœud.
- 2) Une entité extérieure au réseau. Le Sink peut être un dispositif extérieur, par exemple, un ordinateur portable ou un PDA interagissant avec le réseau.
- 3) Une passerelle vers un autre réseau tel qu'Internet, où la demande de l'information vient d'un certain centre de traitement lointain.

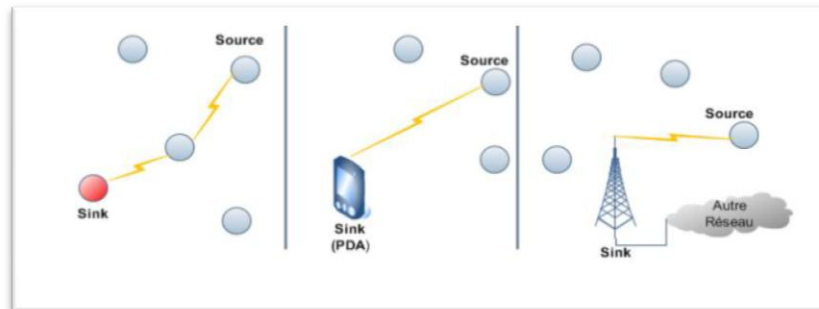


Figure III-9 Les différents types de puits.

III.3.2.3. La pile protocolaire du réseau de capteurs sans fil

La figure III-10 illustre la pile protocolaire du réseau de capteurs sans fil, elle contient cinq couches qui sont : application, transport, réseau, liaison de données et physique, et elle contient aussi trois niveau de gestion : de tâche, de mobilité et d'énergie.

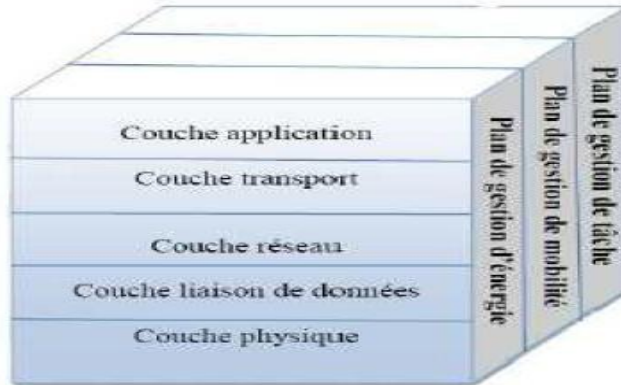


Figure III-10 La pile protocolaire du réseau de capteur.

A. Les couches du RCSF : (ZOUINKHI, 2011)

1) La couche application Elle permet de rendre transparent les mécanismes de communication dans les couches inférieures, en offrant des interfaces pour la création et la diffusion de requêtes. Selon les activités surveillées, différents types d'applications peuvent être construites et utilisées dans la couche application. Elle contient les protocoles suivants :

- Le Protocole de gestion des capteurs (SMP : Sensor Management Protocol)
- Le protocole d'affectation des tâches et d'avertissement des données (TADAP : Task Assignment and Data Advertisement Protocol)
- Le protocole de dissémination des données (SQDDP : Sensor Query and Data Dissemination Protocol)

2) La Couche Transport

- transporter le flux de données à la couche application
- gérer le flux de données si le réseau de capteurs l'exige.
- segmenter les données issues de la couche application.
- réordonne et rassemble les segments venus de la couche réseau
- On utilise le protocole UDP dans la communication entre le nœud collecteur appelé nœud puits (ou sink) et les nœuds de capteurs,

3) La couche Réseau

- l'acheminement des données fournies par la couche transport
- la possibilité de communiquer avec d'autres réseaux de capteurs.

Protocole de couche réseau	Description
SMECN	Crée un sous graphe du réseau de capteurs contenant le chemin donnant le minimum de consommation d'énergie.
SPIN	N'envoie les données aux nœuds que lorsqu'ils sont intéressés. Il a trois types de messages ADV ; REQ et DATA
LEACH	Forme des groupements minimisant la dissipation d'énergie.
Directe diffusion	Initialise des gradients pour le donnée pour flow de la source vers le puits durant une phase de dissémination.

Tableau III-1 Exemples des protocoles de la couche réseaux

4) Couche liaison

- responsable du multiplexage des flux des données,
- du contrôle d'accès au media
- Du contrôle des erreurs
- Le protocole « Machiavel » permet au nœud mobile d'envoyer ses données en étant assuré que ses voisins ont été correctement synchronisés avec réduisant les délais d'accès au médium, et aussi permet également au nœud de ne pas saturer sa file d'attente.

5) Couche Physique

- responsable du support acheminant les données envoyées entre les nœuds.
- il existe deux types de médias pouvant être utilisés pour les réseaux de capteurs: les ondes infrarouges et les ondes radiofréquences

B. Les niveaux de gestion du RCSF : (Khelladi, 2004)

1) Le niveau de gestion des tâches

- L'équilibrage et la distribution des tâches
- Assurer un travail coopératif et efficace

2) Le niveau de gestion de mobilité

- détecte et enregistre tout les mouvements des nœuds capteurs,
- garder continuellement une route vers l'utilisateur final,
- maintenir une image récente sur les nœuds voisins,

3) Le niveau de gestion d'énergie

- gérer l'énergie consommée par les capteurs,

III.3.2.4. Topologie du réseau de capteurs sans fil

Il existe plusieurs topologies pour les réseaux de capteurs (XUE Yong, 2010):

1) Topologie en étoile La topologie en étoile est un système uni-saut. Tous les nœuds envoient et reçoivent seulement des données avec la station de base. Cette topologie est simple et elle demande une faible consommation d'énergie, mais la station de base est vulnérable et la distance entre les nœuds et la station est limitée.

2) Topologie en toile (Mesh Network) La topologie en toile est un système multi-saut. La communication entre les nœuds et la station de base est possible. Chaque nœud a plusieurs chemins pour envoyer des données. Cette topologie a plus de possibilités de passer à l'échelle du réseau, avec redondance et tolérance aux fautes, mais elle demande une consommation d'énergie plus importante.

3) Topologie hybride La topologie hybride est une combinaison des deux topologies ci-dessus. Les stations de base forment une topologie en toile et les nœuds autour d'elles sont en topologie étoile. Elle assure la minimisation d'énergie dans les réseaux de capteurs.

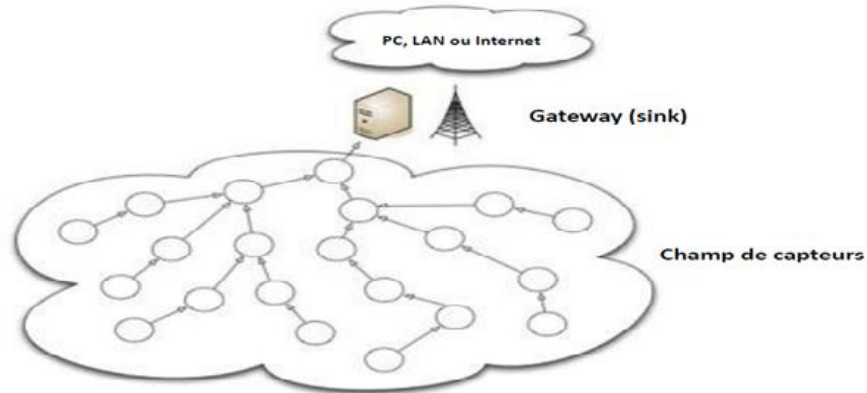


Figure III-11 Topologie hybride d'un réseau de capteurs sans fil.

III.3.3. Technologies de la communication dans les réseaux de capteurs sans fil

Parmi les technologies de la communication dans les réseaux de capteurs sans fil, nous citons les suivantes (XUE Yong, 2010) :

a) Bluetooth / IEEE 802.15.4 Bluetooth est une spécification de l'industrie des télécommunications. Elle utilise une technique radio courte distance destinée à simplifier les connexions entre les appareils électroniques. Malheureusement, un grand défaut de cette technologie est sa trop grande consommation d'énergie.

b) ZigBee / IEEE 802.15.4 ZigBee est une norme de transmission de données sans fil permettant la communication de machine à machine. Zigbee offre des débits de données moindres, mais sa très faible consommation électrique et ses coûts de production très bas en font une candidate idéale pour la domotique ou les matériels de type capteur, télécommande ou équipement de contrôle dans le secteur industriel.

c) Dash 7 / ISO/IEC 18000-7 Dash7 est une nouvelle technologie de réseaux de capteurs sans fil utilisant la norme ISO/IEC 18000-7. Sa consommation électrique est très faible, la durée de vie de batterie peut arriver à plusieurs années. Sa distance de communication est 2km. Elle fournit une faible latence pour le suivi des objets en mouvement, un protocole petite pile, des supports de capteurs et de sécurité et un débit de transmission allant jusqu'à 200kbits/s.

III.3.4. Types de réseaux de capteurs sans fil

Les RCSFs peuvent être classés selon deux points de vue (lamine, 2007):

1) Le model dynamique de réseau Soit le réseau est constitué d'un ensemble de capteurs mobiles évoluant dans un environnement statique. Le but de tels réseaux est la plupart du temps l'exploration de zones inaccessibles ou dangereuses. Les travaux de recherche sont souvent orientés robotique, les nœuds jouant à la fois le rôle de capteur et d'actionneur.

Soit le réseau est constitué de capteurs fixes servant à la surveillance d'occurrence d'événements sur une zone géographique. Ici, le réseau n'effectue que la surveillance, les données mesurées sont transmises en mode multi-sauts à un nœud puits qui est chargé, après réception, de mettre en œuvre les actions nécessaires. Ce puits peut être connecté, de manière filaire par exemple, à un autre réseau.

2) Le model de délivrance de données Soit les capteurs transmettent périodiquement les informations recueillies (délivrance continué de données). Soit le capteur transmet des informations à la détection d'un évènement (délivrance de données basée évènement). C'est à l'utilisateur de lancer une requête pour avoir l'information (en anglais "observerinitiated"). Délivrance de données hybride où on trouve les différentes délivrances, citées avant, en même temps.

III.3.5. Objectif de base du réseau de capteurs sans fil

Les objectifs de base des réseaux de capteurs sans fil dépendent généralement des applications, cependant les tâches suivantes sont communes à plusieurs applications (lamine, 2007) :

Déterminer les valeurs de quelques paramètres suivant une situation donnée. Par exemple, dans un réseau environnemental, on peut chercher à connaître la température, la pression atmosphérique, la quantité de la lumière du soleil, et l'humidité relative dans un nombre de sites, etc.

Détecter l'occurrence des événements dont on est intéressé et estimer les paramètres des événements détectés. Dans les réseaux de contrôle de trafic, on peut vouloir détecter le mouvement de véhicules à travers une intersection et estimer la vitesse et la direction du véhicule.

Classifier l'objet détecté. Dans un réseau de trafic, un véhicule est-il une voiture, un bus, etc.

III.3.6. Caractéristiques des réseaux de capteurs sans fil

Un réseau de capteurs présente les caractéristiques suivantes:

1) L'auto-configuration des nœuds capteurs : Dans un RCSF, les nœuds sont déployés soit d'une manière aléatoire (missile, avion...), soit placés nœud par nœud par un humain ou un robot, et ceci à l'intérieur ou autour du phénomène observé (champ de guerre, surface volcanique, patient malade...) (Akyildiz, 2002). Ainsi, un nœud capteur doit avoir des capacités d'une part, pour s'auto-configurer dans le réseau, et d'autre part pour collaborer avec les autres nœuds dans le but de reconfigurer dynamiquement le réseau en cas de changement de topologie du réseau (Karl, 2003). Dans un RCSF, chaque nœud X possède une unité émettrice/réceptrice qui lui permet de communiquer avec les nœuds qui lui sont proches; En échangeant des informations avec ces derniers, le nœud X pourra alors découvrir ses nœuds voisins et ainsi connaître la méthode de routage qu'il va adopter selon les besoins de l'application (Culler D., 2004).

L'auto-configuration apparaît comme une caractéristique nécessaire dans le cas des RCSF étant donné que d'une part, leur déploiement s'effectue d'une manière aléatoire dans la majorité des applications, et d'autre part le nombre des nœuds capteurs est très grand.

2) Taille importante : un réseau de capteurs peut contenir des milliers de nœuds (BEYDOUN, 2009).

3) La tolérance aux pannes : Dans le cas de dysfonctionnement d'un nœud (manque d'énergie, interférences avec l'environnement d'observation...) ou aussi en cas d'ajout de nouveaux nœuds capteurs dans le réseau, ce nœud doit continuer à fonctionner normalement sans interruption (Akyildiz, 2002). Ceci explique le fait qu'un RCSF n'adopte pas de topologie fixe mais plutôt dynamique.

4) Topologie dynamique : les capteurs peuvent être attachés à des objets mobiles qui se déplacent d'une façon libre et arbitraire rendant ainsi la topologie du réseau fréquemment changeante (BEYDOUN, 2009).

5) Les types de communication : Il existe différents types de communication utilisée dans les RCSF :

a) Unicast : ce type de communication est utilisé pour échanger des informations entre deux nœuds sur le réseau.

b) Broadcast : la station de base ou « Sink » transmet des informations vers tous les nœuds du réseau. Ces informations peuvent être des requêtes de données bien précises (ex : la température dans la région A), des mises à jour de programmes ou des paquets de contrôle... (Akyildiz, 2002)

c) Local Gossip : ce type de communication est utilisé par des nœuds situés dans une région bien déterminée qui collaborent ensemble afin d'avoir une meilleure estimation de l'évènement observé et d'éviter l'émission du même message vers le nœud « Sink » ce qui contribue à consommer moins d'énergie.

d) Convergecast : il est utilisé dans les communications entre un groupe de nœuds et un nœud bien spécifique (qui peut être le « Sink »). L'avantage de ce type de communication est la diminution de contrôle d'entête des paquets (« control overhead ») ce qui économise l'énergie au niveau du nœud récepteur (Karl, 2003).

e) Multicast : il permet une communication entre un nœud et un groupe de nœuds. Ce type de communication est utilisé dans les protocoles qui incluent le « clustering » dans lesquels, le « Clusterhead » s'intéresse à communiquer avec un groupe de nœuds (Akyildiz, 2002).

6) Une architecture « data-centric » : Du fait que le remplacement ou la recharge des batteries des nœuds capteurs est une tâche non pratique et difficile à réaliser, alors il est d'usage normal qu'on trouve des nœuds capteurs redondants (effectuant la même tâche dans la même région) ; L'importance d'un nœud particulier est, par conséquent, réduite par rapport à l'importance attribuée aux données observées par les nœuds. Ce type d'architecture diffère des architectures « node-centric » adoptées par les réseaux traditionnels où les nœuds possèdent une place importante (Exemple : un utilisateur qui veut connecter *son laptop* au *serveur web X*) (Karl, 2003).

7) Une collaboration entre les nœuds : Les contraintes strictes de consommation d'énergie mènent les nœuds capteurs à détecter et traiter les données d'une manière coopérative afin d'éviter le traitement redondant d'une même donnée observée, source de la perte d'énergie.

8) Groupement « clustering »: Les nœuds qui sont regroupés ensemble dans un cluster seront capables de communiquer facilement les uns avec les autres. Les nœuds sont organisés en une hiérarchie en fonction de leur puissance et de leur proximité (Makhoul, 2008).

9) Communication multi-saut: (Makhoul, 2008) Un nœud capteur peut communiquer directement avec ses voisins, c'est-à-dire ceux qui sont à sa portée de communication, et fait office de routeur pour les autres nœuds. On pourrait donc penser que ce mode est plus gourmand en énergie. et de limiter les risques de collisions entre les communications.

10) Mobilité: (Makhoul, 2008) dans un réseau de capteurs mobile les nœuds, la zone d'intérêt, et l'observateur sont mobiles. Comment un nœud peut-il être mobile ? Il

peut être attaché à un robot, à un être humain, véhicule, des animaux, etc. L'objectif des réseaux de capteurs mobiles est de recueillir plus d'informations sur un environnement en utilisant moins de nœuds capteurs, et de permettre au réseau d'organiser lui-même ses nœuds. En outre, il devient capable de déplacer ses capteurs dynamiquement selon les changements environnementaux, ce qui le rend adaptable à l'évolution de son environnement.

11) Identification: Les nœuds dans un réseau de capteurs sont souvent nommés par leurs attributs (Makhoul, 2008).

III.3.7. Défis d'un système de réseau de capteurs sans fil



Dans le tableau suivant, un résumé des défis qu'on rencontre dans la conception et le déploiement d'un système de réseau de capteurs et les solutions proposées (Poslad, 2009) :

Défis d'un système de réseau de capteur	Solutions envisageables
Énergie du capteur constitue une ressource limitée pour la transmission des données	-Utiliser un réseau de capteurs qui se déploie, de faible puissance, à court transmissions. -utilisation des réseaux de capteur multi hop. -Filtrer les données sur place et de transmettre uniquement des données filtrées
La distribution des événements Spatio-temporel dynamique et non déterministe. Peut ne pas être en mesure avant de déterminer comment déployer de façon optimale les capteurs individuels.	-Utiliser un réseau de capteurs pour augmenter la densité de capteur autour des positions estimées de la source du signal quand déterministe; Distributions de capteur de conception pour être reconfigurable, auto-organisation, d'être mobile Soutenez échantillonnage variable et charge les données en rafales collection
Défaillance du capteur est commune en raison d'un manque de puissance, dommage physique, actif (blocage) ou passive environnemental interférence de l'émetteur-récepteur,	Utiliser les réseaux de capteurs denses de faible puissance avec des chemins redondants pour acheminer les données à travers le réseau
Réseaux de capteurs multi-hop peuvent avoir une dynamique topologie. Aucune connaissance globale sur la structure de réseau.	Utiliser des protocoles de routage spécialisé pour travailler sur topologies dynamiques
Les capteurs peuvent être trop coûteux de mettre à jour une fois déployé	conception des capteurs et le nœud de capteur d'accès avec faible maintenance. soutien des capteurs de redondance
Les capteurs peuvent générer d'énormes quantités de données	Utilisation du traitement de données à la fois dans le capteur et le nœud de capteur d'accès

Tableau III-2 Les défis d'un système réseaux de capteur.

III.3.8. Classification des applications des réseaux de capteurs sans fil

La classification par domaines d'application montre le rôle important que peuvent ou pourraient jouer les RCSF. Si l'on considère les applications présentées, on peut définir deux catégories (DE De Sousa, 2008):

-  Les applications d'acquisition de données ;
-  Les applications en remplacement d'une infrastructure de réseau filaire.

La première catégorie contient l'ensemble des applications dont le but est la collecte d'informations. Ces informations proviennent de l'environnement, d'un champ de bataille, d'une machine, ... Ces applications peuvent se découper en trois phases :

- ✓ La phase d'acquisition ;
- ✓ La phase de stockage ;
- ✓ La phase de transmission.

La phase d'acquisition est réalisée par les dispositifs qui mesurent les grandeurs physiques observées. Les phases de stockage et de transmission sont de la responsabilité du capteur sans fil. Le stockage peut être requis par l'application ou peut être nécessaire en cas d'indisponibilité du lien de communication sans fil.

Les RCSF peuvent venir en complément ou remplacement des réseaux filaires. Suite à une catastrophe naturelle comme un tremblement de terre où l'ensemble des infrastructures filaires ont été gravement endommagées, les secouristes pourraient utiliser les RCSF pour communiquer entre eux et mieux s'organiser.

Dans un musée ou un monument historique où l'on souhaiterait disposer de commodités tel qu'Internet, l'utilisation des capteurs sans fil aurait l'avantage de ne pas nécessiter de lourds travaux d'infrastructure demandant des autorisations parfois difficiles à obtenir.

III.4. Relation entre RCSF et les systèmes sensibles au contexte

Les systèmes sensibles au contexte (CAS : context awareness system) sont capables d'adapter leurs comportements en basant sur les informations reçues à partir de leurs contextes. Les fournisseurs des informations du contexte sont les capteurs. Les applications conscientes au contexte sont formées de deux parties. La première, la partie intérieure, est l'application lui-même qui peut adapter son fonctionnement en basant sur les informations fournies par la deuxième partie. La deuxième, la partie extérieure, est responsable de rassembler les informations à partir de l'environnement, de les traiter et de les présenter à la partie intérieure.

Le RCSF fournit une infrastructure de base pour rassembler les données contextuelles. Ces données sont souvent en forme brute, inadapté pour les applications, ils décrivent seulement les phénomènes physiques détectés directement. Le RCSF permet l'accès à des informations contextuelles fiables et de façon permanente. Il fournit aussi des niveaux d'abstraction différents des informations contextuelles.

A. Exigences de l'application sensible au contexte Pour qu'un RCSF puisse soutenir les applications sensibles au contexte, directement, exactement et faiblement, il doit satisfaire les exigences demandées par les applications sensibles au contexte qui sont les suivantes (Tahayori Hooman, 2007) :

1) Séparation des préoccupations une séparation claire, entre la partie intérieure et la partie extérieure de l'application sensible au contexte, permet de présenter des interfaces claires à toutes les applications sensibles au contexte pour qu'elles soient connectées au RCSF quelconque existant dans la région.

2) Communication transparente et distribué les informations contextuelles nécessaires pour une application sensible au contexte ne sont pas trouvées, souvent, dans le même domaine, mais sont recueillies auprès de différents champs de capteurs. En cas où, la machine exécutant l'application est présente dans un seul endroit, un nœud spécial,

nommé agrégateur qui est responsable de connecter les différents réseaux de capteurs, est proposé pour assurer la disponibilité de toutes les informations contextuelles.

3) Disponibilité constante de l'information le RCSF fournit en permanence des informations contextuelles. Afin de réduire la charge dans le réseau et l'ordinateur exécutant l'application, l'application doit être en mesure d'enregistrer ses besoins dans des agrégateurs ou / et des puits liés selon le niveau dont il a besoin pour travailler. Bien que tout changement se produise dans l'environnement intéressant pour l'application, puits ou agrégateur doivent informer l'application.

4) Stockage et Historique des contextes RCSF doit être capable d'enregistrer tout ce qu'il est détecté, qui peut être utilisé soit par des applications ou même par le réseau. Grâce à l'historique du contexte, le réseau peut prédire son futur contexte et même estimer son taux d'erreur ou de ses fonctionnalités. En cas d'une erreur très grande, le réseau demande à l'application de changer le réseau si c'est possible.

5) Découverte des ressources Comme les applications sensibles au contexte ne sont pas conscientes de toutes les ressources disponibles, il doit y avoir une carte de route disponible pour trouver les ressources nécessaires. Alors que dans une zone, en étant conscient de l'existence de capteurs, l'application doit essayer de trouver le puits ou agrégateur. Dans le puits, toutes les ressources disponibles doivent être enregistré et être offert à l'application à sa demande.

6) Fiabilité d'informations En tout ESN un mécanisme pour le calibrage de réseau est nécessaire. Etant donné que le calibrage de chaque nœud de capteur n'est pratiquement pas possible, le calibrage doit être fait de manière passive dans les nœuds de niveau supérieur. Chacun des nœuds puits doit compenser l'erreur. Le Macro Calibration est un procédé de calibration du RCSF qui est également applicable. RCSF peut traiter des informations inexacts par trois façons. En passant l'information erronée à l'application en indiquant que l'information n'est pas exacte, ou de tenter de retirer l'erreur, ou demander à l'utilisateur de gérer manuellement l'erreur.

B. La relation entre RCSF et CAS Les réseaux de capteurs sans fil sont liés aux systèmes sensibles au contexte, au moins, dans deux façons. Premièrement, ces systèmes utilisent des capteurs pour acquérir des informations contextuelles. Deuxièmement, les capteurs eux-mêmes peuvent être amenés à prendre conscience de leur propre contexte. (LOKE, 2006).

Dans l'informatique pervasif, les applications sont supposées être mobiles et autonome. Et comme la sensibilité au contexte se base sur les capteurs qui fournissent les informations contextuelles, ces applications ne doivent pas être liées à des infrastructures fixes. Par conséquent, le RCSF est la solution adéquate pour assurer la mobilité et la disponibilité des informations contextuelles aux applications sensibles au contexte. Nous déduisons que les RCSF sont la base des applications sensibles au contexte, comme l'illustre la figure III-12.

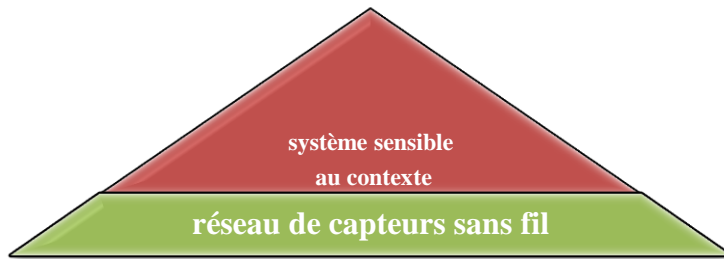


Figure III-12 relation entre RCSF et système sensible au contexte

III.5. Conclusion

Nous avons vu dans ce chapitre les réseaux sans fil de manière générale et les réseaux de capteurs sans fil de façon plus détaillée, nous avons présenté les capteurs sans fil avec leurs caractéristiques et leur architecture. Puis nous avons étudié tout ce qui concerne les réseaux de capteurs sans fil, à savoir son architecture, ses caractéristiques, sa classification, et ses technologies. Nous avons montré la relation entre les RCSF et les systèmes sensibles au contexte.

Nous déduisons que les RCSF sont d'une grande importance dans le domaine d'acquisition des informations, et ceci grâce aux grands développements au niveau des capteurs et au niveau de la conception des RCSF, nous déduisons aussi, après cette étude, que la plupart des applications des RCSF sont des applications sensibles au contexte mais l'inverse est faux. Les RCSF jouent un rôle important dans les systèmes sensibles au contexte.

Dans le chapitre suivant nous mettons en valeur cette relation à travers les travaux existants qui relient les RCSF et les systèmes sensibles au contexte.

CHAPITRE IV. TRAVAUX CONNEXES

IV.1. Introduction

Les applications, les objets ou les systèmes sensibles au contexte sont conscients de leur environnement et les circonstances peuvent répondre intelligemment. En outre, les nouvelles technologies de capteurs et réseaux de capteur sans fil fournissent l'intelligence de l'environnement et la capacité de capter, de raisonner et d'agir sur l'environnement. Cependant, avec la prolifération des réseaux de capteurs sans fil, il y a un intérêt croissant dans les systèmes sensibles au contexte qui utilisent des facteurs contextuels (physiques) externes telles que la localisation, la présence, la température et la lumière des informations et d'interagir avec l'environnement.

L'intégration des réseaux de capteurs à un niveau mondial est une étape essentielle sur le chemin d'un couplage plus étroit entre le monde virtuel et le monde réel. Dans un système sensible au contexte avec des milliers voire des millions d'éléments de données décrivant la situation actuelle, il est évident que ces données doivent être filtrées et agrégées afin de distribuer la détection et le traitement efficace. La nécessité d'agréger des données existe aussi du point de vue du réseau de capteurs sans fil puisque l'énergie est limitée et il existe un besoin fort de réduire le trafic de communication.

Après avoir présenté les systèmes sensibles au contexte et les réseaux de capteurs sans fil, nous verrons dans le chapitre suivant les travaux existants qui relient ces deux concepts, nous aborderons la classification et la description de ces travaux.

IV.2. Classification des Systèmes Sensibles au Contexte

Les réseaux de capteurs sont liés à l'informatique sensible au contexte au moins dans deux façons. Au premier lieu, on utilise des capteurs pour acquérir des informations contextuelles pour les entités. D'autre part, les capteurs eux-mêmes peuvent être amenés à être conscients de leur propre contexte. (LOKE, 2006). Les travaux en relation avec notre projet peuvent alors être classés en deux classes :

IV.2.1. Réseaux de Capteurs Sensibles au Contexte

Dans cette classe le réseau de capteurs est sensible à son contexte, de tel sorte qu'il peut changer son comportement selon les informations tirées du son contexte. Par exemple la gestion de l'énergie des nœuds, en cas où l'énergie d'un nœud atteint la limite minimum (information du contexte), le réseau peut réagir d'une façon favorable.

IV.2.2. Systèmes Sensibles au Contexte à base des Réseaux de Capteurs

Ici le réseau de capteurs est utilisé dans un autre système comme un moyen d'acquisition des informations contextuelles Systèmes. Donc ce système peut réagir selon les informations contextuelles fournis par le réseau de capteurs. Une application sensible au contexte est formée de deux parties :

A) Partie un : l'application lui-même qui adapte ses fonctionnalités selon les changements du contexte.

B) **Partie deux** : responsable de rassembler les informations du l'environnement.

IV.3. Exemples de Travaux de chaque Classe

IV.3.1. Réseaux de Capteurs sensibles au contexte

IV.3.1.1. A Context-Aware Approach to Conserving Energy in Wireless Sensor Networks

Dans cet article (Chong Suan Khai, 2005), les auteurs proposent l'idée des senseurs conscients du contexte qui adaptent leurs états du fonctionnement selon leur contexte. Leur approche est d'appliquer le concept de la sensibilité au contexte pour la réduction de l'énergie. Ils présentent un Framework pour supporter l'utilisation du contexte pour provoquer les fonctionnalités d'économie d'énergie des senseurs. D'après l'application ils ont affirmé que le contexte rassemblé des données des senseurs est une alternative faisable pour la gestion de l'énergie des senseurs.

IV.3.1.2. Context-Aware Sensors .

(Elnahrawy Eiman, 2004) Les auteurs introduisent les capteurs conscients du contexte, ils proposent une technique de modélisation et d'apprentissage de l'information contextuelle statique dans le réseau de capteurs, cette information permet aux senseurs de prédire leurs lectures actuelles en basant sur leurs anciens lectures et les lectures actuelles de leurs voisins.

Cette technique est basée sur le classificateur Bayesian.ils planifient le problème de l'apprentissage et l'utilisation de l'information contextuelle selon le problème de l'apprentissage des paramètres d'un classificateur Bayesian, puis faire des inférences, respectivement. Les résultats trouvés montrent que cette approche fonctionne bien dans les données du senseur avec des caractéristiques spatio-temporelles.

IV.3.1.3. Context-Aware Sensornet.

(Huaifeng Qin, 2005) Les réseaux de capteurs (sensornet) déployés dans des zones hostiles et éloignées doivent, eux même, exécuté des ajustements adaptatifs.

Dans ce contexte les auteurs présentent l'idée du réseau de capteurs conscient du contexte(CASN), les nœuds de ce réseau sont prévus d'utiliser le contexte disponible et d'ajuster leurs comportements selon le contexte. Aussi ces nœuds utilisent le contexte pour promouvoir leur collaboration.

Ce travail présente la motivation et le défi d'établir un réseau de capteur conscient du contexte, les problèmes d'implémentation du middleware du CASN et les technologies autorisée du middleware des CASN.

IV.3.2.Sensibles au Contexte à base des Réseaux de Capteurs.

IV.3.2.1. Enhanced sensor network:

A specialized infrastructure for context-aware applications. (Hooman Tahayori, 2007) : cet article a présenté un cadre pour l'intégration des réseaux de capteurs selon les besoins des application sensible au contexte, et pour améliorer le réseau de capteurs sans fil comme une infrastructure qui peut fournir les informations contextuelles

nécessaires pour les applications sensibles au contexte. Le projet de cadre se traduira par un réseau hiérarchique à partir d'au moins deux points de vue :

- Un, comme un réseau tel qu'il est,
- L'autre en tant que fournisseur d'informations hiérarchiques.

ENS (Enhanced sensor network) est un réseau de capteurs spécialisés qui donne accès à l'information contextuelle exigée à partir d'une interface uniforme pour toute application sensible au contexte, qui s'exécute effectivement ou virtuellement dans l'environnement. ENS, en raison de sa nature peut utiliser de capteurs matériel et software, et peuvent être installés dans tout type d'environnement, réel ou virtuel.

Le RCSF doit satisfaire les exigences suivantes pour supporter les systèmes sensibles au contexte :

- 1) **Séparation** : entre les deux parties de l'application sensible au contexte.
- 2) **Communication transparente et distribué** : Un nœud spécial est proposé, nommé agrégateur qui est responsable de connecter les réseaux de capteurs différents. Ce nœud est capable d'effectuer un niveau d'abstractions plus élevée.
- 3) **Disponibilité constante de l'information** : ESN fournit en permanence des informations contextuelles, mais cela est dû à l'application comment les utiliser
- 4) **Stockage et Historique des contextes** : ESN doit être capable d'enregistrer tout ce qu'il est détecté qui peut être utilisé soit par des applications ou même par le réseau.
- 5) **Découverte des ressources** : Comme les applications sensibles au contexte ne sont pas conscientes de toutes les ressources disponibles, il doit y avoir une table de route disponible pour trouver leurs ESN et les ressources nécessaires.
- 6) **Fiabilité d'informations** : En tout ESN un mécanisme pour le calibrage de réseau est nécessaire. Etant donné que le calibrage de chaque nœud de capteur n'est pratiquement pas possible, le calibrage doit être fait de manière passive dans les nœuds de niveau supérieur.

IV.3.2.2. SAIL

a Sensor Abstraction and Integration Layer for Context Awareness L'auteur de (Girolami Michele, 2008) a proposé une architecture de couche appelée SAIL (sensor Abstraction and intégration layer) à être utilisés dans les architectures conscients au contexte, et visant à intégrer le RCSF comme sources d'information de contexte. SAIL simplifie l'acquisition de données et l'interaction entre les applications du RCSF et les systèmes sensibles au contexte existant.

Dans l'approche proposée, les applications fonctionnant sur RCSF peuvent être exposés à l'aide soit une centrée sur le nœud ou un paradigme centré de données, et ils sont interfacés avec des technologies d'accès différentes. SAIL est encapsulé dans le Platform OSGi et a été testé sur mottes MICAz.

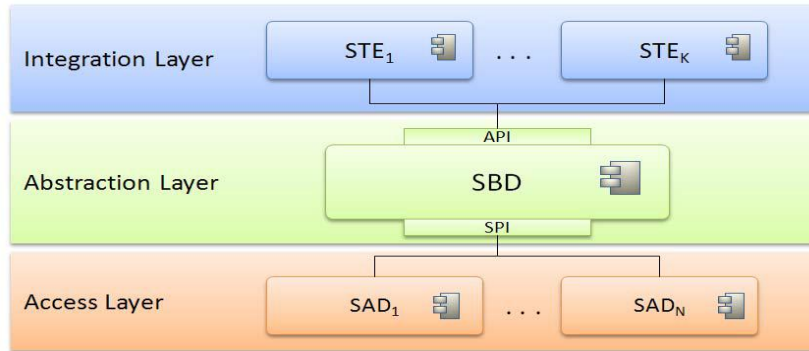


Figure IV-1 l'architecture des couches SAIL

IV.3.2.3. CAS²

Context Awareness Service System based on Wireless Sensor Networks Les auteurs de cet article (Park Ilgon, 2006) ont proposé l'architecture du système (CAS²) pour le traitement des informations de contexte en basant sur le réseau de capteurs sans fil, CAS² peut fournir des services divers et pratiques à l'utilisateur avec seulement des actions minimales en collectant des informations de contexte de l'utilisateur en temps réel.

CAS² fournit également des services optimisés pour chaque utilisateur sur la base des informations de contexte. Dans cet article, ils ont présenté une synthèse sur le système CAS² avec différents éléments.

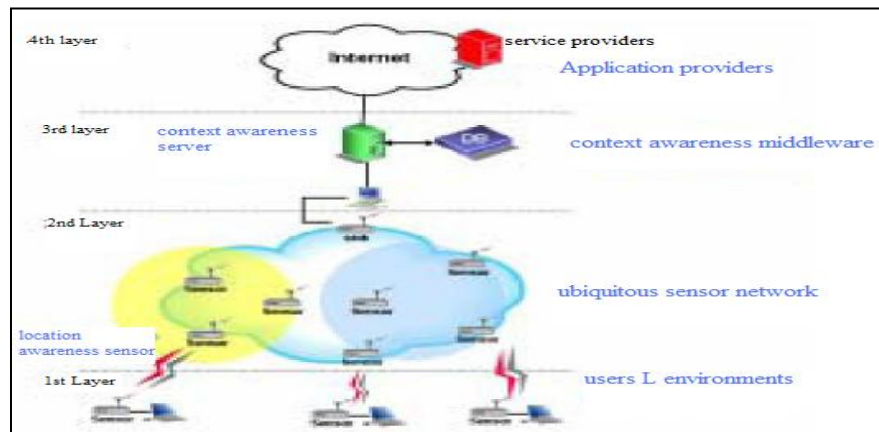


Figure IV-2 Architecture des systèmes sensible au contexte.

IV.3.2.4. A Context Aware Motor Controller Based on WSN

Zhicheng Deng et Ming Zhu (Deng, et al., 2010) , a leur avis, les développements récents dans les domaines de la détection, communication et l'informatique ont conduit les efforts de recherche aux dispositifs sensibles au contexte, qui peuvent utiliser des informations de contexte pour réaliser une interaction plus intuitive entre l'homme et la machine.

Les auteurs ont élaboré une nouvelle conception de systèmes intelligents Contrôleur de moteur qui est capable d'accéder à des capteurs d'environnement et acquérir des informations connexes, de manière à utiliser une commande différente

régimes et des paramètres de contrôle pour répondre à différentes tâches et environnements. Cette application consciente au contexte a besoin d'un nouveau cadre pour la collecte d'information du contexte et le traitement correspondant.

Le cadre proposé dans ce travail se compose de trois composantes principales: une carte intelligente ZigBee, réseaux de capteurs sans fil et un panneau configurable. Le RCSF déployé dans l'espace physique fournit l'information de l'environnement et le panneau configurable est responsable de remplir des fonctions configurables de commande du moteur.

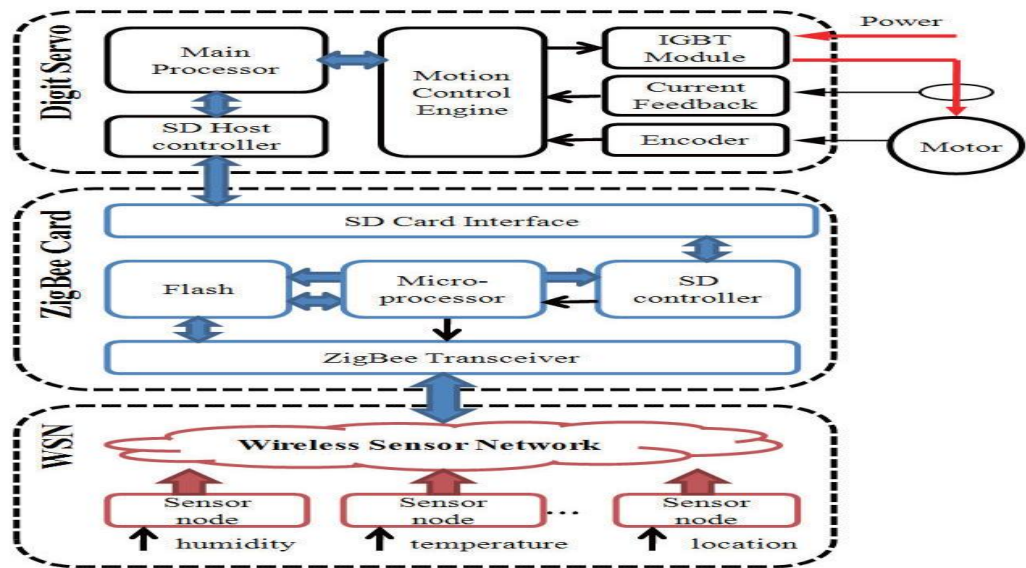


Figure IV-3 Framework du moteur contrôleur sensible au contexte.

IV.3.2.5. CONSORTS-S

A Mobile Sensing Platform for Context-Aware Services. Dans Cet article (Sashima Akio, 2008) les auteurs ont proposé une plate-forme de détection mobile qui fournit des services sensibles au contexte pour les utilisateurs mobiles en accédant aux réseaux de capteurs sans fil à l'entour. La plate-forme fournit les facilités suivantes:

- 1) communiquer aux RCSF via un routeur de capteur mobile attaché à un utilisateur de téléphone mobile.
- 2) analyse des données détectées issues de réseaux en coopérant avec capteur middleware sur un ordinateur distant serveur pour capturer ses contextes.
- 3) fournir les services sensibles au contexte pour les utilisateurs mobiles de téléphones cellulaires.

La plate-forme est conçue pour traiter les données détectées efficacement grâce à la coopération entre un téléphone mobile, un capteur mobile routeur, et un intergiciel de capteur sur un serveur distant.

paramètre est analysé séparément, elles conduisent souvent à la génération de fausses alarmes et on limite ainsi l'acceptabilité de ces systèmes.

Ils ont proposé non seulement de combiner les données des multi-capteurs disponibles sur les appareils-Wearable, mais aussi pour interfacer les nœuds portables avec un réseau maillé de dispositifs de détection déployés dans l'environnement.

Une telle solution permet une analyse contextuelle de données physiologiques pour une évaluation correcte de la situation et la génération d'alarme fiable.

Le réseau de capteurs sans fil fournit une alimentation fiable et un peu de moyen de communication pour les appareils-Wearable. Leur système basé sur les données de l'ECG et d'accélération acquis par les nœuds portables avec la localisation descriptive et les données de l'environnement à partir du réseau de capteurs sans fil.

Ainsi ils ont présenté l'architecture du système proposé et un exemple d'implémentation à la fois à l'intérieur et à l'extérieur. Le système proposé est facile à mettre en œuvre, flexible et évolutive qui le rend approprié pour le déploiement dans les grandes surfaces.

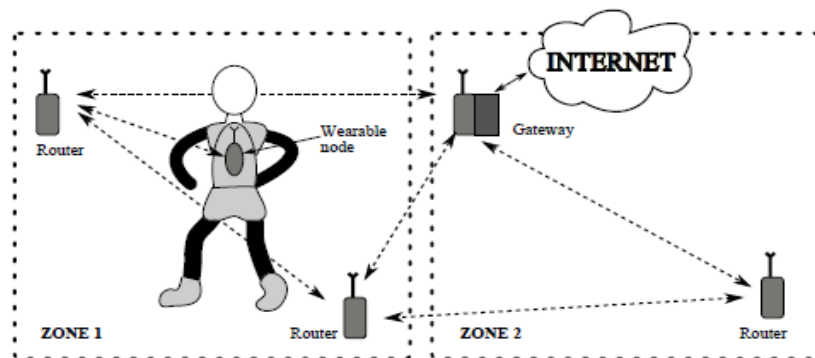


Figure IV-6 Représentation d'un système avec un nœud portable.

IV.3.2.7. Study on the Context-Aware Middleware for Ubiquitous Greenhouses Using Wireless Sensor Networks.

Les auteurs de l'article (Hwang, 2011) considèrent que le réseau de capteurs sans fil est l'une des technologies importantes pour mettre en œuvre la société de l'ubiquité, et il peut augmenter la productivité des produits agricoles et d'élevage. Middleware, qui peut connecter le matériel de RCSF, les applications et les systèmes d'entreprise, est nécessaire pour construire l'environnement de l'agriculture omniprésente combinant la technologie de RCSF avec des applications dans le secteur agricole.

Cet article propose un middleware sensible au contexte pour traiter efficacement les données recueillies auprès de serres omniprésentes en appliquant la technologie de RCSF ; ce middleware est utilisé pour mettre en œuvre des services combinés grâce à la connectivité organique de données.

Le middleware proposé abstrait les nœuds de capteurs hétérogènes pour intégrer des données de différentes formes, et fournit des fonctions sensibles au contexte, service

d'événements, et de filtrage intelligent pour optimiser l'opérabilité et l'évolutivité du middleware.

Pour évaluer la performance de middleware, un système de gestion intégré pour les serres omniprésentes a été mis en œuvre par l'application de middleware proposé à une serre existante, et il a été testé en mesurant le niveau de charge à travers l'utilisation du processeur et le temps de réponse pour les demandes des utilisateurs lorsque le système fonctionne.

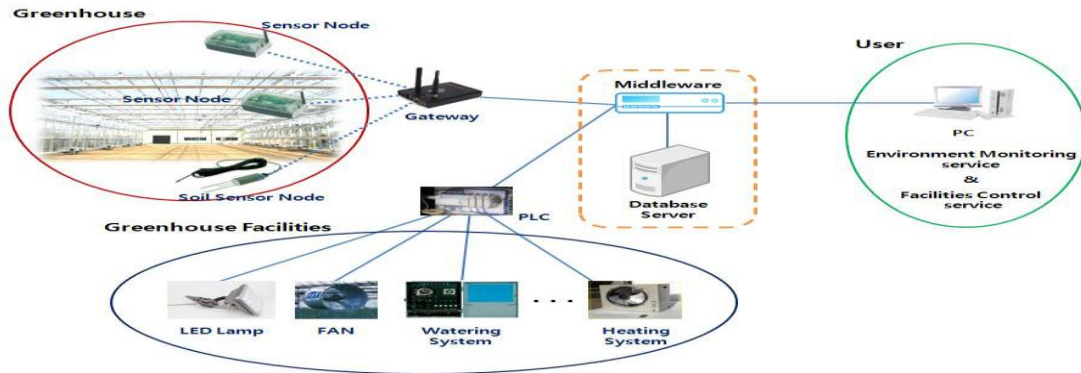


Figure IV-7 la structure du système de la serre omniprésente.

IV.3.2.8. Using Wireless Sensor Networks to Narrow the Gap between Low-Level Information and Context Awareness

Dans cet article, (Raicu Ioan, 2004) les auteurs ont présenté une architecture évolutive d'utilisation des réseaux de capteurs sans fil en combinaison avec réseaux Ethernet sans fil pour fournir une solution complète du bout en bout pour réduire le fossé entre l'information du niveau bas et la sensibilisation de contexte.

Ils ont développé et mis en œuvre un détecteur de proximité complète afin d'envoyer à l'ordinateur portable, tel qu'un PDA, le contexte de la position.

Puisque la position n'est qu'un élément de la conscience de contexte, ainsi ils ont cherché à utiliser des capteurs de photo et les capteurs de température pour apprendre autant que possible sur l'environnement.



Figure IV-8 Architecture de BuL (niveau de construction) et les niveaux de WSN.

IV.4. Conclusion

Nous avons décrit, dans ce chapitre, la classification des applications sensibles au contexte, ces applications sont réparties en deux classes :

- ◆ Réseaux de Capteurs sensibles au contexte.
- ◆ Systèmes Sensibles au contexte à base des réseaux de capteurs.

Sont illustrés les travaux de chaque classe. A partir de cette étude des travaux nous déduisons que les RCSF est une partie essentielle qu'on ne peut soustraire des applications sensibles au contexte. Dans le chapitre suivant nous ferons part de notre contribution.

CHAPITRE V. CONTRIBUTION

V.1. Introduction

La tolérance aux fautes demeure l'une des principales caractéristiques des RCSF : c'est la capacité de gérer les pannes arrivant sans, pour autant, arrêter le fonctionnement du RCSF,

Dans la première partie de ce chapitre, nous étudions les différents types de pannes et les solutions existantes, vu l'importance de cet aspect du RCSF.

La deuxième partie concerne les travaux de base pour notre contribution : après une étude des applications sensibles au contexte. Notre projet repose sur le travail fait par Ahmed ZOUINKHI sous le titre «Contribution à la modélisation de produit actif communicant, spécification et évaluation d'un protocole de communication orienté sécurité des produits » (Zouinkhi, 2011). L'objectif de ce travail se place dans le contexte de l'amélioration de la gestion de la sécurité des produits et des personnes à travers le concept d'objets intelligents appliqué aux produits chimiques dangereux.

Nous basons aussi notre travail sur un deuxième travail se rapportant à la détection de pannes dans les RCSF (Trab Sourour, 2013) , ce travail est appliqué au système présenté dans (Zouinkhi, 2011) .Pour compléter ce travail nous ferons la correction de pannes. Pour cela, nous avons choisi une méthode de détection et de correction de pannes dans les RCSF qui est présenté dans le travail intitulé « A Technique to Identify and Substitute Faulty Nodes in Wireless Sensor Networks » (Taleb, 2009).

La troisième partie explique notre contribution qui consiste à détecter les nœuds en panne dans le RCSF en se basant sur la redondance matérielle et à remplacer ces nœuds.

Nous avons utilisé cette méthode dans l'application choisie qui est celle du contrôle des produits en utilisant un RCSF.

V.2. Partie 1 : La tolérance aux Pannes dans les RCSF

Introduction Certains nœuds capteurs peuvent être bloqués ou tomber en panne à cause d'un manque d'énergie, d'un dégât matériel ou d'une interférence environnementale. La panne d'un nœud capteur ne doit pas affecter le fonctionnement global de son réseau : C'est le problème de fiabilité ou de tolérance aux pannes (Challal, 2008) .

V.2.1. Panne

V.2.1.1. Définition : Faute, erreur et faille

A. **Une faille (ou panne)** du système se produit lorsque son comportement devient inconsistant et ne fournit pas le résultat voulu. La panne est une conséquence d'une ou plusieurs erreurs.

B. **Une erreur** représente un état invalide du système du à une faute (défaut).

C. La **faute** est donc la première cause de l'erreur, cette dernière provoque la faille du système.

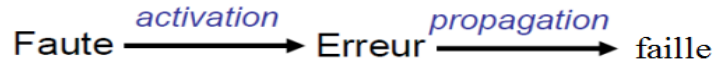


Figure V-1 relation entre faute, erreur et faille

(Challal, 2008)

Selon la figure V-1 on peut aller de la faute à Une faille (ou panne) comme suit : Quand une faute est **activée**, elle provoque une erreur. En **propageant**, l'erreur engendre une faille (Challal, 2008).

V.2.1.2. Causes des Pannes dans les RCSF

On peut distinguer, spécialement pour les réseaux de capteurs, trois principales causes de pannes:

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

V.2.1.3. Classification des Pannes

Il est utile de classer les pannes selon différents critères. Le schéma suivant montre une classification générale selon la durée, la cause ou le comportement d'une panne (Challal, 2008):

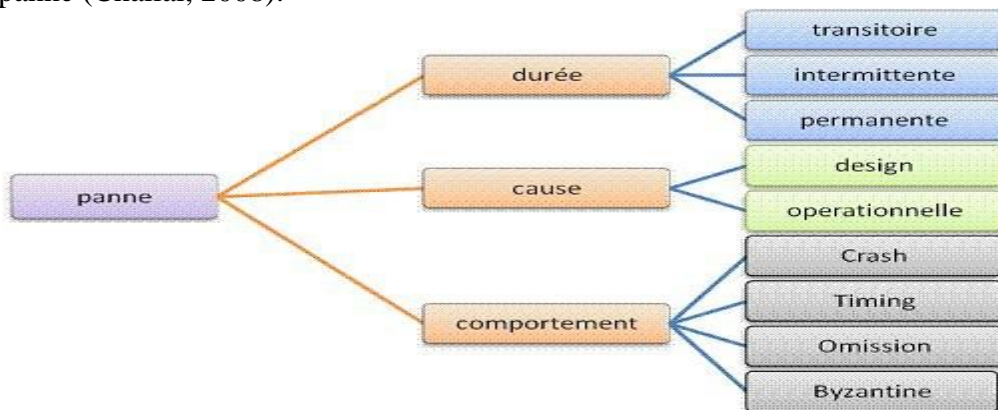


Figure V-2 Classification des pannes

(Challal, 2008)

1) Les pannes selon la durée En basant sur la durée de la panne, on peut classer la panne en :

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

2) Les pannes selon la cause On distingue deux types de pannes selon leurs causes

- **Panne de design** : due à une mauvaise structuration du réseau ou du composant en particulier. En pratique, ce genre de panne ne devrait pas exister grâce aux tests et simulations avant la réalisation finale du réseau ;
- **Panne opérationnelle** : qui se produit durant le fonctionnement du système. Elle est généralement due aux causes physiques.

3) Pannes selon le comportement résultant. Après l'occurrence d'une panne, on distingue quatre différents comportements possibles du composant concerné (Challal, 2008):

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

V.2.2. Tolérance aux Pannes

V.2.2.1. Définition

La tolérance aux pannes est la capacité de maintenir les fonctionnalités du réseau sans interruption due à une panne d'un nœud capteur.

Le but de la tolérance aux pannes est d'éviter la faille totale du système malgré la présence de fautes dans un sous ensemble de ses composants élémentaires.

V.2.2.2. Procédure de tolérance aux pannes

La conception d'une procédure pour la tolérance aux pannes dépend de l'architecture et des fonctionnalités du système. Cependant, certaines étapes générales sont exécutées dans la plupart des systèmes; tel que c'est illustré dans la figure V-3 (Challal, 2008) :

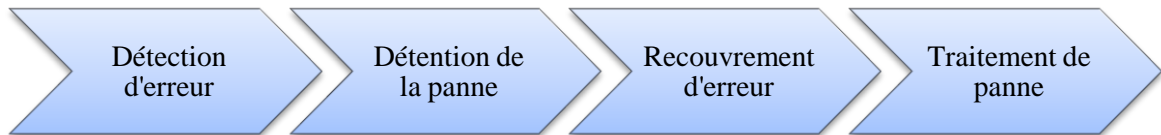


Figure V-3 Procédure de tolérance aux pannes

1) Détection de l'erreur :

C'est la première phase dans chaque schéma de tolérance aux pannes, dans laquelle on reconnaît qu'un événement inattendu s'est produit. Les techniques de détection de pannes sont généralement classifiées en deux catégories : en ligne et autonome (offline). :

A. La détection offline est souvent réalisée à l'aide de programmes de diagnostic qui s'exécutent quand le système est inactif.

B. La détection en ligne vise l'identification de pannes en temps réel et est effectuée simultanément avec l'activité du système.

2) Détection de la panne :

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

3) Recouvrement d'erreur :

C'est la phase dans laquelle on effectue des opérations d'élimination des effets de pannes. Les deux techniques les plus utilisées sont le masquage de la panne et la répétition :

A. Masquage de panne : utilise l'information redondante correcte pour éliminer l'impact de l'information erronée ;

B. Répétition : après que la panne soit détectée, on effectue un nouvel essai pour exécuter une partie du programme, dans l'espoir que la panne soit transitoire.

4) Traitement de panne :

Dans cette phase, la réparation du composant en panne isolé est effectuée. La procédure de réparation dépend du type de la panne. Les pannes permanentes exigent une substitution du composant avec un autre composant fonctionnel. Le système doit contenir un ensemble d'éléments redondants (ou en état standby) qui servent à remplacer les nœuds en panne.

V.2.3. Classification des solutions de tolérance aux pannes dans les RCSF

Le but des techniques de tolérance aux fautes est de limiter les effets d'une faute, ce qui signifie d'augmenter la probabilité qu'une erreur soit acceptée ou tolérée par le système. Une caractéristique commune de toutes les techniques de tolérance aux fautes est l'utilisation de la redondance. La redondance est tout simplement l'ajout de ressources matérielles ou temporelles au-delà de ce qui est nécessaire pour un fonctionnement normal du système. Il peut s'agir (AMIN, 2011) :

- de matériel : certains modules matériels sont dupliqués, ou tripliqués.
- de temps : des parties d'un programme sont exécutées plusieurs fois,
- de l'information : le circuit ou le programme utilise une information redondante,
- ou un mélange des trois solutions précédentes.

Dans la suite nous décrivons les différentes classes des techniques de tolérance aux fautes existantes:

V.2.3.1. Classification selon la détection d'erreurs

Il y a trois classes de techniques de détection d'erreurs: redondance matérielle, redondance temporelle, et redondance d'informations (AMIN, 2011).

1) **Redondance matérielle** : La redondance matérielle est l'approche couramment utilisée. Elle fait référence à l'ajout de ressources matérielles supplémentaires, tel que le doublement du système, en utilisant un comparateur en sortie pour détecter les erreurs.

Cette technique peut être divisée en deux sous-types : la duplication avec comparaison et la duplication avec redondance complémentaire.

- **La duplication avec comparaison** (Duplication With Comparison, DWC) ou double redondance modulaire (Dual Modular Redundancy, DMR) est une technique de détection d'erreur simple et facile à mettre en œuvre. Elle possède une bonne capacité de détection d'erreur, sauf pour les bogues dus à la conception, les erreurs dans le comparateur, ou les combinaisons d'erreurs simultanées dans les deux modules.

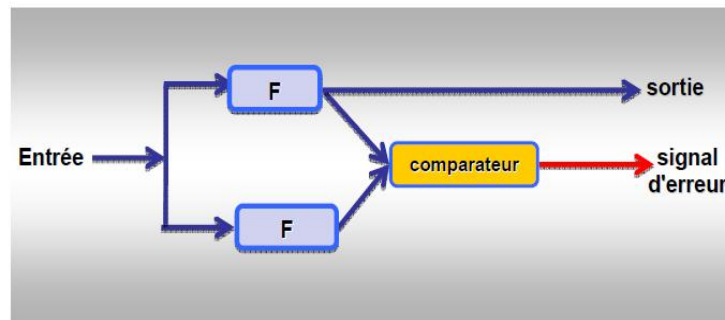


Figure V-4 Duplication avec comparaison

(AMIN, 2011)

- **duplication avec redondance complémentaire** (Duplication with Complement Redundancy, DWCR) : Cette technique est similaire à la DWC, à la différence que les signaux d'entrée, les signaux de contrôle

en sortie, ainsi que les signaux de données internes de chacun des deux modules sont de polarité opposée pour interdire les erreurs simultanées dans les deux modules afin d'éviter la défaillance du système.

2) Redondance temporelle : Elle désigne une technique de redondance qui nécessite une seule unité effectuant une même opération deux fois de suite. Si une différence est constatée entre les deux calculs successifs, cela signifie qu'une faute transitoire ou intermittente est apparue lors de l'un ou l'autre des calculs. Dans cette approche, il y a une pénalité en termes de temps supplémentaire, cependant la pénalité matérielle est moindre. Il s'agit d'une technique de réplication temporelle, sans considération de la fonctionnalité du circuit (AMIN, 2011).

3) Redondance d'information : L'idée sous-jacente d'un schéma de redondance d'information est d'ajouter de l'information redondante aux données transmises, stockées ou traitées, afin de déterminer si des erreurs ont été introduites. C'est une façon de protéger les données grâce à un codage mathématique qui peut être réutilisé ensuite pour décoder les données originales (AMIN, 2011).

V.2.3.2. Classification selon la correction d'erreurs

Un système avec détection d'erreur est un système sûr, mais il ne peut continuer à fonctionner tant qu'il n'a pas la capacité de corriger les erreurs. De manière similaire à la détection des erreurs, les techniques de correction sont aussi classées en trois sous-classes : la redondance matérielle, temporelle ou d'information (AMIN, 2011).

1) Redondance Matérielle : Il existe trois types de techniques (Imad, 2011):

- Les techniques passives
- Les techniques actives ou dynamiques
- Les techniques hybrides.

➤ Redondance matérielle passive :

L'objectif de la redondance matérielle passive est de masquer les fautes et d'éviter ainsi leur propagation en erreurs. Son mécanisme de base est le vote majoritaire. Un exemple est la redondance modulaire triple (Triple Modular Redundancy ou TMR).

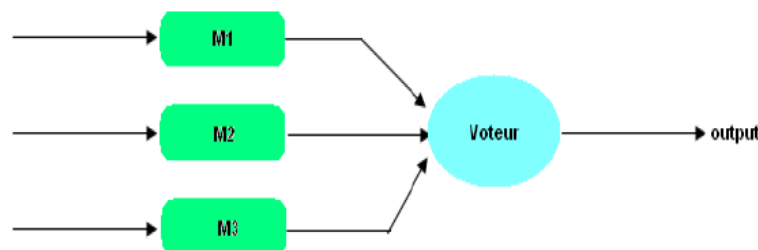


Figure V-5 TMR (triple modular redundancy).

La Figure V-5. Présente le TMR. Tel que : M1, M2 et M3 sont trois modules qui génèrent des données homogènes. Le voteur permet de produire la donnée finale.

Généralisation de TMR

On généralise le cas TMR de la figure V-5 à N modulaires : Redondance N-modulaire (N-modular redundancy ou NMR). N modules identiques sont utilisés avec N est un nombre impair.

Les systèmes utilisant la redondance matérielle passive se caractérisent par les points suivants :

✓ **Niveau d'insertion du vote**

On donne dans la figure V-6 un exemple où Le vote peut être inséré à différents niveaux : Système de contrôle industriel.

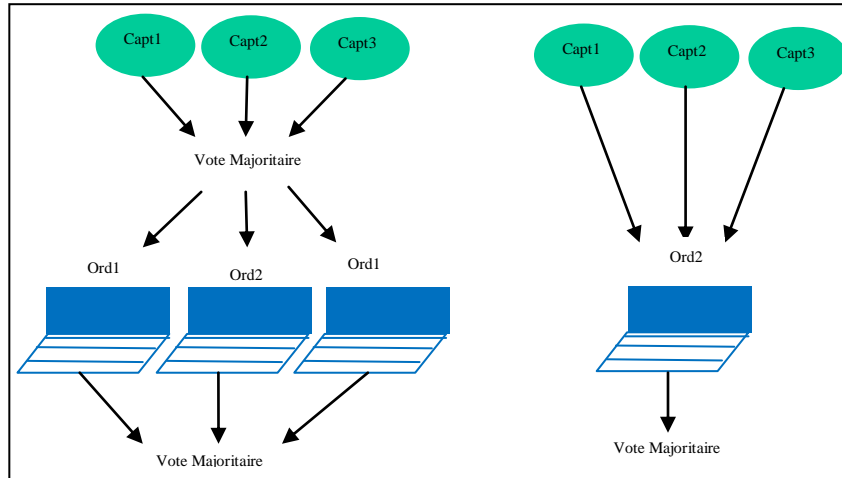


Figure V-6 Système de contrôle industriel.

✓ **Choix de conception**

Le voteur peut être un voteur matériel ou un voteur logiciel.

Les critères pour choisir le type du voteur sont:

- la disponibilité d'un processus,
- la vitesse d'exécution du vote,
- la limitation de puissance, de poids,
- le nombre de voteurs à réaliser.

✓ **Discordance des valeurs obtenues**

C'est quand les valeurs fournies par les différents modules sont différentes alors que l'environnement est correct. Dans ce cas, on considère souvent la valeur médiane.

➤ **Redondance matérielle active :**

Il existe plusieurs types de redondance matérielle active : la duplication et la comparaison, l'utilisation des modules « spares » et l'approche « pair-and-a-spare ».

❖ **Duplication et comparaison**

Elle consiste à :

- Dupliquer le même matériel.
- Exécuter la même application sur chaque instance du matériel.
- Comparer les résultats obtenus.

Les limitations de ce type de redondance sont :

- Le cas d'entrées fausses.
- L'inexactitude de la comparaison.
- La défaillance du comparateur.

❖ **Utilisation des modules « spares »**

On parle dans cette technique du remplacement standby ou de la disponibilité standby. Le principe est qu'un seul module est opérationnel, les autres sont des spares

(modules disponibles) ; si on détecte des fautes dans le module principal, on procède à son remplacement.

On peut avoir :

- Un remplacement à chaud (modules de remplacement sont opérationnels).
- Un remplacement à froid (modules de remplacement non opérationnels).

❖ **L'approche « pair-and-a-spare »**

Dans cette approche deux modules fonctionnent en parallèle et leurs résultats sont comparés. Si un module est défaillant, on le remplace par un spare. On peut remplacer la paire entière en cas de défaillance.

➤ **Redondance matérielle hybride :**

Dans cette technique, on combine la redondance active et la redondance passive. La faute est masquée pour éviter la propagation en erreur. Si une erreur a eu lieu, elle est détectée et localisée pour permettre une reconfiguration du système.

2) Redondance temporelle : Pour la correction d'erreur à l'aide de la redondance temporelle, un calcul est répété sur le même matériel à trois intervalles de temps différents puis un vote intervient sur les résultats. Elle exige donc trois fois plus de cycles d'horloge pour exécuter la même tâche.

La redondance temporelle peut uniquement corriger les erreurs dues aux fautes transitoires à condition que la durée de la faute soit inférieure au temps de calcul. Ayant besoin de temps supplémentaire pour répéter les calculs, elle ne peut être employée que dans les systèmes avec peu ou pas de contraintes temporelles (AMIN, 2011).

V.3. Partie 2 : Les Travaux de base de mon projet

Introduction Après l'étude de l'état de l'art des travaux qui relient les applications sensibles au contexte et les RCSF, nous avons choisi un travail qui est un système sensible au contexte à base du RCSF (Zouinkhi, 2011) et un deuxième travail qui traite la détection de pannes dans les RCSF (Trab Sourour, 2013), ce travail est appliqué au système présenté dans (Zouinkhi, 2011). Pour compléter ce travail nous allons faire la correction de pannes dans RCSF par la méthode décrite dans (Taleb, 2009).

V.3.1. Travail 1 : Système sensible au contexte à base du réseau de capteurs sans fil

V.3.1.1. Introduction

Mon travail intitulé « Contribution des réseaux de capteurs à la conscience des contextes des systèmes ». Nous avons alors fait notre recherche sur les applications sensibles au contexte basées sur les réseaux de capteur sans fil.

Après une étude de ces applications nous avons choisi de baser notre projet sur le travail fait par Ahmed ZOUINKHI sous le titre «Contribution à la modélisation de produit actif communicant, spécification et évaluation d'un protocole de communication orienté sécurité des produits » (Zouinkhi, 2011).

L'objectif de ce travail se place dans le contexte de l'amélioration de la gestion de la sécurité des produits et des personnes à travers du concept d'objets intelligents appliqué aux produits chimiques dangereux (Zouinkhi, 2011). Il s'agit d'analyser, de définir, de formaliser et de mettre en place des mécanismes d'intelligence ambiante dans les environnements de stockage et de logistique industrielle dans le domaine chimique, afin d'améliorer les interactions entre les produits et les acteurs du système de manière à assurer la sécurité active des biens et des personnes (Zouinkhi, 2011). Le produit peut être acteur, et contrôleur de son usage, sa maintenance, son stockage, sa sécurité, son transport, ...

V.3.1.2. Problématique du Sujet de Recherche

La problématique de recherche concerne la modélisation du comportement interne d'un objet communicant et de ses interactions ainsi que les processus de coopération entre objets communicants. Le contexte est un environnement à intelligence ambiante, intégrant des produits et objets physiques équipés d'intelligence embarquée et de moyen de communication sans fil (RFID, Réseau de capteurs, ...) dans l'objectif d'assurer la sécurité intrinsèque et extrinsèque des produits industriels à caractère chimique dangereux (Zouinkhi, 2011).

V.3.1.3. Concept de Produit Actif

Le concept de produit dit actif consiste à doter un produit de capacités à communiquer, informer, acquérir, décider et réagir aux stimuli et perturbations de son environnement afin de permettre au produit de s'adapter, d'influer, de coopérer, de

transformer le comportement de son environnement. Le produit est ainsi un acteur intelligent et proactif dans son environnement ambiant avec lequel il interagit au moyen de communication sans fil.

Dans l'industrie chimique, on peut utiliser ce concept sur un produit industriel de type conteneur pour l'administration de sécurité des biens et des personnes.

Cet objet est composé d'un fût contenant une substance chimique auquel est attaché un dispositif microélectronique qui peut communiquer avec d'autres dispositifs qui sont attachés aux autres fûts.

Un Produit à Intelligence Ambiante est capable de « sentir » son environnement, à travers des capteurs embarqués, de décider et faire un choix d'action/réaction selon des spécifications propres et/ou partager dans un environnement coopératif, de communiquer avec son environnement.

Un produit est un objet qui joue deux rôles : un rôle « passif » de capture de mesure (des valeurs ambiantes) et un rôle « actif » de action/réaction (réaliser des échanges et des interactions intelligentes avec les autres produits) à travers des capteurs embarqués (Zouinkhi, 2011).

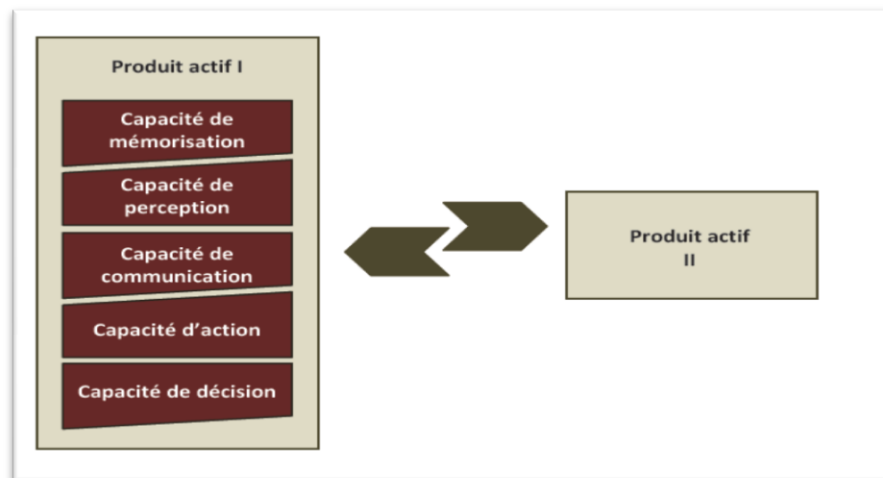


Figure V-7 les capacités d'un produit actif

(Zouinkhi, 2011)

V.3.1.4. Modèle de Produit Actif Proposé par Zouinkhi

Ce modèle fonctionnel proposé du produit actif est représenté dans la figure V- 8 . Il se compose de plusieurs rubriques, sont les suivants (Zouinkhi, 2011):

a) Un bloc de gestionnaire des dispositifs perception Qui est chargé comme interface entre la base de connaissance et les capteurs et actionneurs.

b) Une base de connaissances Qui transforme les informations perçues par ses propres capteurs et par celles des voisinages en action de sécurité relativement à ses règles décisionnelles qui sont de trois types : statique (ambiantes), dynamique et communautaire (voir tableau V-1).

Domaine de connaissance	Produit (Type Produit, ID, Symbole)
Règles statiques	D=Température (<LimInf ou >LimSup) B=Température (in [LimInf + Δ, LimSup - Δ]) M=Température (in [LimInf, LimInf + Δ])
Règles dynamiques	Danger=Délai (Mauvais)>Période critique
Règles de communauté	Distance (PA, X) Symbole Produit % Table de compatibilité
Actionneurs	Message= « Incompatible & Distance » ou « M » Alerte= « Incompatible & Distance<Dmin » ou « D »

Tableau V-1 base des connaissances du produit actif

c) **Un bloc de services** Dans lequel on trouve les blocs d'inscription qui est lié aux informations de la base de connaissance.

d) **Un module de surveillance de voisinage** : Qui collecte les informations des voisinages et les transmet à la base de connaissance pour être traitées (Zouinkhi, 2011)

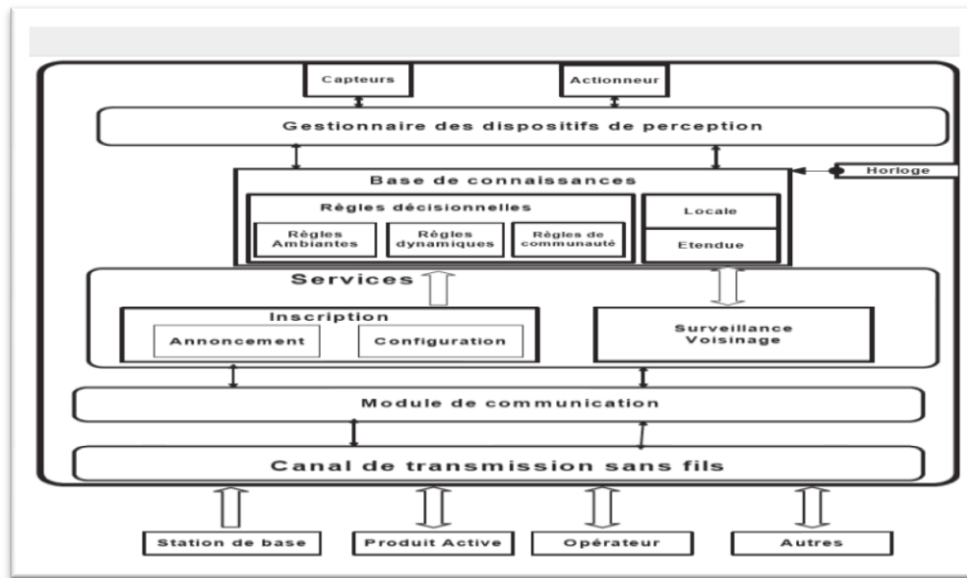


Figure V-8 modèle fonctionnel de produit actif

(Zouinkhi, 2011)

V.3.1.5. Tâches du Produit Actif

Le produit actif peut effectuer deux tâches essentielles pour son bien être (Zouinkhi, 2011):

A. Surveillance Interne Où il est capable de surveiller son voisinage. La violation des règles de sécurité individuelles ou mutuelles doit être détectée et analysée. La base de connaissance de produit est composée des faits et des règles.

B. Surveillance et la Communication avec d'autres Produits Actifs Cette tâche se fait par le message de salutation GRE qui est transmis automatiquement et périodiquement entre les produits dans leur état normal de fonctionnement.

Le message GRE porte les informations propres du produit (nom, symboles de sécurité, son niveau de sécurité actuel,...), et a pour rôle ultérieurement de contribuer au processus de calcul de la distance séparant deux produits actifs.

Dès qu'un produit reçoit un message GRE, il va émettre un Message RSI : L'information de ce type de message contient principalement la différence de puissance du signal. Cette technique de localisation s'appelle RSSI (Received Signal Strength Indicator) et se base sur une équivalence entre la valeur de la puissance du signal et la valeur de la distance séparant les deux produits.

V.3.1.6. Comportement du Produit Actif

Les produits actifs coopèrent ensemble par échanges d'informations sur l'environnement et calculent son niveau de sécurité en temps réel, par conséquent un produit actif peut gérer sa sécurité et la sécurité de son environnement.

Le mécanisme de coopération intègre deux approches : l'une centralisée côté administration et l'autre décentralisée coté coopération entre produits actifs.

Pour passer vers l'état Actif le produit doit subir une stratégie, il passe obligatoirement dans des états précis à savoir l'inscription, la configuration, la surveillance et la communication et finalement la surveillance interne.

Le comportement du produit actif est illustré dans la figure V-9 :

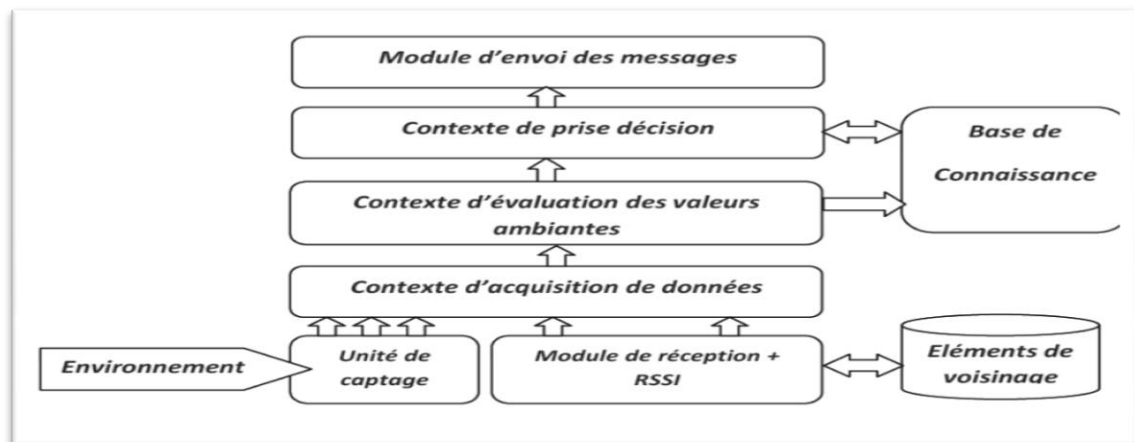


Figure V-9 comportement autonome du produit actif

(Zouinkhi, 2011)

1) Annonce du produit dans la communauté L'étape d'inscription est décisive pour qu'un nouveau produit s'introduise dans une communauté. L'inscription se fait chez le gestionnaire pour que ce dernier le détecte et l'ajoute dans sa liste des produits déjà existants.

2) Configuration Chaque élément de la communauté doit avoir une liste de paramètres pour qu'il soit identifié. Cette liste se compose de deux groupes : les paramètres de configuration générale et les règles de sécurité.

3) Surveillance et Communication Le produit actif, doit interagir avec son environnement. Il doit être sensible à toutes modifications de son environnement qui violent les règles de sécurité individuelles ou réciproques, pour qu'il réagisse en temps réel en transmettant ces anomalies vers le superviseur. Le superviseur peut demander à des produits la lecture de certains paramètres tels que les paramètres de configuration ou les valeurs ambiantes lues par les capteurs associés.

4) Surveillance interne La surveillance est la procédure où le produit actif applique ses règles ambiantes et commence à traiter les mesures générées depuis sa plateforme pour en décider finalement le niveau de sécurité courant.

Les messages qui peuvent émettre dans ce cas sont les messages d'alerte annonçant un état de sécurité menaçant, par exemple un état où la distance entre deux produits est dépassée, et nécessitant une intervention immédiate. Ce message reporte au superviseur l'état de sécurité défaillant avec les mesures qui ont provoqué cet état.

La situation d'alerte est subdivisée en deux catégories :

➤ ALER généré s'il y a incompatibilité entre produits actifs proches :

Si la distance entre deux produits est inférieur à une limite critique, le produit qui a détecté le premier ce dépassement déclenche une alerte.

➤ ALEV reporte l'état déficient des mesures des valeurs ambiantes :

A part le fonctionnement normal du système, il y a des cas exceptionnels où le produit se trouve dans un état de dysfonctionnement. Dans ce cas, le produit doit envoyer un message d'alerte ALE en diffusion afin d'avertir les autres qu'il y a un problème dans ce produit. Le produit reste bloqué en attente du message ACKALE de la part du superviseur. Ce dernier n'envoie ce message que lorsque le problème est résolu. (Zouinkhi, 2011)

V.3.1.7. Scénario de Fonctionnement Complet du Réseau

Dans le fonctionnement normal du réseau et dans le cas favorable qui ne correspond à aucune alerte, les produits configurés entrent en communication entre eux selon le modèle du scénario Salutation Entre Produits. Et chacun réalise sa fonction principale, qui est la détection périodique de la valeur du processus contrôlé, selon le scénario Capture des Valeurs Ambiantes.

Après l'étape de configuration, le produit entre dans une boucle définie par deux activités qui sont :

- la salutation entre les produits.
- la capture des valeurs ambiantes.

Il y a d'autres activités qui se déclenchent de façon événementielle suivant les besoins du superviseur comme :

- la demande de lecture des valeurs ambiantes,

- la demande de lecture des paramètres de configuration,
- la demande de lecture des règles de sécurité. (Zouinkhi, 2011)

V.3.1.8. Messages Echangés dans le Réseau

Tableau V-2 suivant illustre les différentes natures des messages émis ainsi que leurs désignations, ces messages sont présentés en couple requête/réponse.

Requête		Réponse	
Nom	Désignation	Nom	Désignation
CTR	Demande d'inscription d'un produit actif	Ack_CTR	Acquittement du gestionnaire
NCF0	Demande suite à un manque des règles et des symboles de sécurité	CMD1+CMD3	
NCF1	Demande suite à un manque des règles de sécurité	CMD3	Règles de sécurité appropriées envoyé par le gestionnaire
NCF2	Demande suite à un manque des symboles de sécurité	CMD1	Symboles de sécurité appropriée envoyé par le gestionnaire
GRE	Salutation entre produits actifs	RSI	Puissance de signal reçu équivalent à la distance qui sépare les Produits actifs
CMD2	Demande de gestionnaire sur l'état de configuration d'un produit actif	CFG	Réponse d'un produit actif annonçant sa configuration
CMD4	Demande de gestionnaire des règles de sécurité d'un produit actif	SER	Réponse d'un produit actif annonçant ses règles de sécurité
CMD5	Demande de gestionnaire des variables ambiantes d'un produit actif	INA	Réponse d'un produit actif contenant ses variables ambiantes
Rapp_D	Rapport état danger vers le gestionnaire	Ack_Rapp_D	Acquittement sur état Danger renvoyé par le gestionnaire
Rapp_M	Rapport état mauvais vers le gestionnaire	Ack_Rapp_M	Acquittement sur état mauvais renvoyé par le gestionnaire

Tableau V-2 Ensemble des messages échangés requête/réponses

(Zouinkhi, 2011)

V.3.1.9. Conclusion et Critique

Dans ce travail l'auteur a utilisé le réseau de capteur sans fil pour gérer la sécurité des produits chimiques dangereux. Les produits deviennent des entités intelligentes qui communiquent et font des réactions. L'application est sensible au contexte grâce aux capteurs collés aux produits. Les produits génèrent des alertes lorsqu'il y a des événements anormaux tels que l'incompatibilité entre produits actifs proches et l'état déficient des mesures des valeurs ambiantes.

Critique Dans le travail (Zouinkhi, 2011) l'auteur n'a pas étudié les pannes des produits. Cependant, pour assurer bien la sécurité des produits chimiques la vérification de l'état de fonctionnement des produits doit être assurée à tout moment. Si un produit est non fonctionnel (il ne fait aucune action notamment la communication), le superviseur doit détecter ce produit et régler la panne. Dans ce sens que nous avons choisi de faire notre travail sur la détection de la panne dans le travail de (ZOUINKHI, 2011)

V.3.2. Travail 2 : Détection de pannes dans les RCSF

V.3.2.1. Introduction

L'auteur a étudié la détection de pannes dans son réseau dans l'article intitulé: "Application of distributed fault detection in WSN to dangerous chemical products based on Bayesian approach" (Trab Sourour, 2013). Dans son article, l'auteur a étudié la méthode de détection des fautes distribuée dans le réseau de capteurs en utilisant l'approche Bayésienne; il a réalisé une application pour les produits chimiques dangereux. Ce travail vise à résoudre le problème de détection des fautes dans le réseau de capteurs distribués.

V.3.2.2. Approche Bayésienne

C'est la méthode pionnière appliquée dans l'industrie des produits chimiques dangereux pour le pré-déploiement des nœuds capteurs collés aux produits. Cette approche fournit la tolérance aux fautes et améliore la détection des fautes (Trab Sourour, 2013).

V.3.2.3. Principe de détection de fautes

La méthode de détection de fautes adoptée par l'auteur consiste en deux phases (Trab Sourour, 2013):

Phase une : Chaque nœud prend une décision locale sur la présence d'un événement rare selon les valeurs captées par les capteurs et envoie cette décision à la station de base.

Phase deux : La station de base prend la décision finale en se basant sur une règle optimale de fusion des décisions reçues.

V.3.2.4. Conclusion et Critique

L'auteur a traité le problème de détection de pannes, il a traité la présence ou l'absence des fautes et il n'a pas pris en considération la correction de la faute.

Cependant, la correction de la faute est un facteur principal qui influence sur la sécurité des produits chimiques, surtout en cas d'arrêt total du nœud, c'est pour cette raison que nous nous intéressons à étudier la détection et la correction de la panne dans le système de (Zouinkhi, 2011).

V.3.3. Travail 3: Détection et correction de pannes dans les RCSF

V.3.3.1. Introduction

Le travail choisi est intitulé « A Technique to Identify and Substitute Faulty Nodes in Wireless Sensor Networks ». (Taleb, 2009)

Dans son article, l'auteur présente une technique pour identifier les nœuds défectueux et la substitution de ces nœuds pour atteindre la tolérance aux pannes dans les réseaux de capteurs sans fil. La technique proposée divise le réseau en zones disjointes, tout en ayant un maître pour chaque zone. Les maîtres de la zone sont utilisés pour identifier les nœuds défectueux par la division de la zone en quadrants jusqu'à ce qu'un nœud suspect soit trouvé. Ce modèle de détection de fautes suppose à la fois la communication et la détection de défauts qui sont causés par une défaillance matérielle dans un nœud. En outre, l'auteur a utilisé une nouvelle technique pour activer les nœuds en

sommeil afin de valider la correction des nœuds suspects. Cette technique est utilisée pour reconfigurer le réseau en utilisant uniquement des nœuds non défectueux. Lorsqu'on parle de la tolérance aux fautes dans RCSF on prend en considération deux types de fautes (Taleb, 2009):

- Nœud défectueux sans fournir des données.
- Nœud actif en fournissant des données incorrectes.

V.3.3.2. Technique Proposée

Elle consiste à diviser le réseau en des zones disjointes et chaque zone a son maître, lorsque une faute est détectée le maître est responsable d'identifier le nœud suspect ; il va éveiller les nœuds dormis et selon la réaction des nœuds le maître peut décider si le nœud suspect est défectueux ou non.

Architecture du réseau Le réseau est composé des nœuds homogènes et la plupart d'eux sont actifs alors que ceux en état dormis ne sont pas nombreux. Les nœuds sont statiques. Le réseau est devisé en quatre zones principales, dans chaque zone le nœud actif, avec un ID le plus grand, est choisi comme le maître de la zone.

V.3.3.3. Types de Pannes Traitées

Les fautes prises en considération, dans le travail de (Taleb, 2009), sont celles de sensation et de communication causées par la défaillance matérielle.

1) Détection de fautes de communication Elle consiste à :

- ☞ définir un seuil pour le débit de communication (τ).
- ☞ mesurer le débit de communication T dans la zone de teste.
- ☞ comparer les deux valeurs pour décider s'il y a de faute ou non. Si ($T < \tau$) alors une faute est détectée.

2) Détection de fautes de sensation Elle consiste à calculer l'écart entre les déférentes lectures des senseurs.tel que les données envoyées par un nœud seront comparés avec les valeurs de la plages des données stocké déjà chez le maître de la zone ; si les données rapportées s'écartent des valeurs stockées alors, le maître déduit qu'il y a une faute de sensation dans le réseau.

V.3.3.4. Aperçu de la technique

A. Calculer périodiquement le débit de communication :

- 1) Chaque maître de zone calcule son débit de communication et le compare avec le seuil.
- 2) s'il y a un faute, il va deviser sa zone en quarts virtuels.
- 3) Il calcule le débit de communication pour chaque quart et le compare avec un autre seuil.
- 4) Si une faute est détectée dans une quart, il devise ce quart en quatre autre quarts virtuels.
- 5) Il continue la division en quarts virtuels et le calcul du débit de communication jusqu'à trouver un quart qui contient un seul nœud.

- 6) Il peut alors identifier que ce nœud est suspect.
- 7) Puis il va éveiller les nœuds voisins dormis pour tester le nœud suspect.

B. Tester si les données sont correctes :

- Chaque nœud envoie des données au maître de la zone.
- Le maître de la zone compare ces données avec le statut du nœud et l'intervalle des valeurs enregistré dans le maître de la zone.
- Si les données envoyées écartent de celles enregistrées alors il y a une faute.
- Le maître de la zone va diviser la zone en quarts virtuels jusqu'à l'identification du nœud suspect.
- Puis il va éveiller les nœuds voisins dormis pour tester le nœud suspect.

C. Phases de la Technique

1) Initialisation :

- ❖ Les nœuds sont groupés en quatre zones principales selon leurs positions.
 - ❖ Dans chaque zone, le nœud actif avec le grand ID (identificateur) est choisi comme maître de la zone.
 - ❖ Le maître de la zone enregistre la trace des données lui envoyées par les nœuds de sa zone.
 - ❖ Le maître de la zone est le responsable d'envoyer ces données à la station de base.
 - ❖ Le maître de la zone garde la trace des données reçues et les informations des nœuds de sa zone en utilisant deux tables : table des informations et table des enregistrements.
- a. Table des informations** Elle contient l'ID du nœud envoyant le message, la taille du message et une marque du temps qui indique le temps d'arriver du message.

NodeID	Message_length	Time_Stamp
8	4	0.501
1	4	0.504
6	4	1.002
2	4	1.003

Tableau V-3 Table des informations

- b. Table des enregistrements** Elle contient les ID des nœuds de la même zone et leurs positions. Elle permet au maître de la zone de garder la trace des nœuds dans sa zone et aussi de diviser la zone en quarts.

NodeID	XPosition	YPosition	XMax	YMax	Center
1	23	5	30	30	15
2	16	13	30	30	15
10	7	7	30	30	15
11	16	17	30	30	15

Tableau V-4 Table des enregistrements

- ❖ Pour la détection des fautes, le maître de la zone maintenir **une table grille**, il divise sa zone en une grille et enregistre les informations de sa zone dans cette table grille (voir table 8) :

Square-Number	Enclosed_nodes	Low	Medium	High
1	[9, 10]	0	0	1
4	3	0	1	0
7	[4, 13]	1	0	0
9	5	1	0	0

Tableau V-5 table grille

2) Détection de fautes Pour décider si le nœud suspect est défectueux on applique la technique à base de la redondance dans le réseau.

A. La foute de communication :

- Le maître de la zone cherche le nœud dormi le plus proche du nœud suspect.
- Après l'identification de ce nœud il va l'éveiller pour commencer à sentir.
- Après un temps le maître va calculer le débit de communication des deux nœuds : le nœud suspect et le nœud éveillé.
- Si : la différence entre les deux valeurs est supérieur d'un seuil Δ Et le débit du nœud suspect est inférieur de celui du nœud éveillé Alors un autre nœud dormi sera éveillé.
- Le troisième nœud va résoudre le conflit du nœud défectueux.
- Il commencera à sentir.
- Après un temps le maître calcule le débit de communication du troisième nœud et la compare à Δ
- Si : les valeurs du débit de communication des nœuds éveillés sont égaux et la différence entre ces deux valeurs et la valeur du débit de communication du nœud suspect est grand, le maître décide que le nœud suspect souffre de faute de communication.
- Le nœud suspect sera mis en état off et l'un des nœuds éveillés restera éveillé et l'autre reviendra à dormir.

B. La faute de sensation :

- On compare les données reçus d'un nœud avec son entrée dans la table grille.
- Si les données reçus ne sont pas inclus dans la rangée correcte et le nœud n'a pas un statut correct, le nœud est considéré comme étant suspect.
- Le maître commence à diviser sa zone en quarts jusqu'à trouver le nœud suspect.
- Pour décider si le nœud suspect est défectueux on procède de la même façon précédente qui consiste à éveiller les nœuds dormis proches du nœud suspect. (Taleb, 2009)

V.3.3.5. Conclusion

La technique proposée, selon les résultats auxquels arrive l'auteur, est valable et réussie pour la détection des nœuds défectueux. Elle permet de détecter plusieurs nœuds défectueux dans la même zone ; aussi elle peut détecter les nœuds défectueux dans plusieurs zones en même temps. La précision de cette technique est très élevée lorsque le nombre des nœuds défectueux est inférieur au nombre de nœuds en sommeil.

V.4. Partie 3 : la Contribution

Introduction Dans cette partie nous, présentons notre contribution qui se base sur le travail intitulé « Contribution à la modélisation de produit actif communicant, spécification et évaluation d'un protocole de communication orienté sécurité des produits » (Zouinkhi, 2011) qui est une application sensible au contexte à base d'un réseau de capteurs sans fil pour le contrôle des produits chimiques dans un entrepôt. Notre contribution est inspirée aussi du travail sous le titre « A Technique to Identify and Substitute Faulty Nodes in Wireless Sensor Networks » (Taleb, 2009), ce travail consiste à détecter exactement le nœud défectueux dans un RCSF et à substituer ce nœud par un autre nœud redondant.

Notre contribution est, pour compléter aussi le travail du (Trab Sourour, 2013) qui fait la détection des pannes dans les RCSF, ce travail est appliqué sur le système de (Zouinkhi, 2011).

V.4.1. Notre Objectif

Notre objectif est de réaliser deux idées principales :

- ☞ Ajouter la notion de redondance des nœuds de capteurs au modèle du produit actif : Appliquer la méthode de redondance matérielle (Spares) pour la correction de pannes dans le RCSF.
- ☞ Détection des nœuds défectueux dans le RCSF et les substituer par des nœuds redondants.
- ☞ Montrer l'influence du RCSF dans le système sensible au contexte à travers sa caractéristique de tolérance aux pannes.

V.4.1. La preuve qu'on a une sensibilité au contexte

La définition d'un système sensible (conscient) au contexte, donnée par Dey et Abowd, (CHAARI, 2007) est : « un système est sensible au contexte s'il utilise le contexte pour fournir des informations et des services pertinents pour l'utilisateur, où la pertinence dépend de la tâche demandée par l'utilisateur », Dans notre cas nous avons :

- Le maître de la zone peut déterminer le nœud en panne (information contextuelle)
- Le maître de la zone fait ensuite une réaction qui est le remplacement du nœud en panne par l'un des nœuds redondants à ce nœud (réaction automatique selon une information contextuelle sans l'intervention de l'utilisateur).
- Un nœud actif déclenche des alertes en cas de détection des informations (contextuelles) incompatibles selon ses règles de compatibilités (comme la distance).

V.4.2. La preuve qu'on a un réseau de capteurs

Les nœuds actifs se communiquent entre eux par des messages (comme les messages de salutations).

V.4.2. Description de la Structure du Réseau

Notre réseau à réaliser est un réseau de capteurs sans fil, il est composé d'un ensemble des produits actifs qui peuvent interagir de façon autonome. A chaque produit actif, on associe un capteur principal en état actif et trois autres capteurs secondaires (redondants) en état dormi (passif). Chaque capteur aura un identifiant et le capteur principal aura un identifiant élevé par rapport aux identifiants des capteurs secondaires. Le réseau contient aussi une station de base qui fait la configuration de tous les produits actifs dans le réseau, et d'autres fonctions. Nous décomposons notre réseau en quatre zones où chaque zone a son maître dont l'identifiant est le plus élevé dans sa zone et qui est un nœud actif.

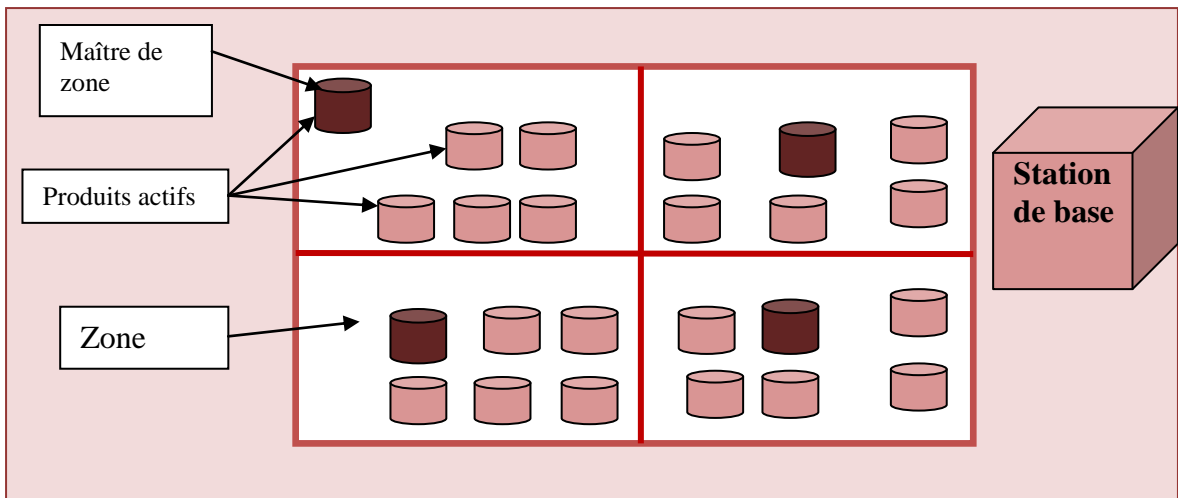


Figure V-10 Structure du réseau

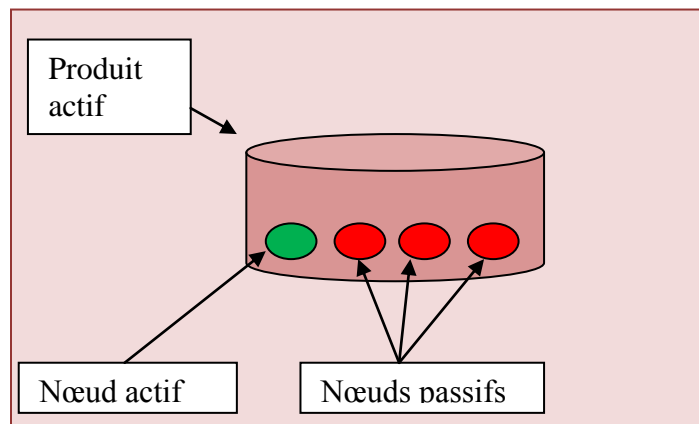


Figure V-11 Forme d'un produit actif

V.4.3. Fonctions de Chaque Composant du Réseau

Composant	fonctions
Station de base	-enregistrement des nœuds dans le réseau dans sa base de données. -récupération des données de tous les nœuds du réseau à partir des maîtres des zones. - responsable de contrôler le fonctionnement des maîtres des zones.
Maître de zone (nœud actif associé à un produit actif)	-rechercher des nœuds en panne. - désactivation des nœuds en panne. -activation des nœuds redondants. -transmettre les données reçues des nœuds de sa zone à la station de base. -enregistrer la trace des nœuds de sa zone (les messages reçus). -vérifier le nombre des nœuds passifs pour chaque produit.
Nœud actif (capteur actif associé à un produit actif)	-faire la configuration de ses nœuds redondants NR (passifs). -envoyer la liste de ses NRs avec leurs configurations à la SB. -envoyer des messages de salutation à ses voisins (nœuds actif) périodiquement -calculer la distance qui le sépare des autres nœuds actifs. -déclencher une alerte en cas de dépassement de la distance. - envoyer les données capturées au maître de la zone.
Nœud passif (capteur en état dormis associé au produit)	-rester en état passif (aucune fonction) jusqu'à qu'il devient en état actif par l'intervention du maître de la zone.

Tableau V-6 Fonctions des composants du réseau.

V.4.4. Caractéristiques du réseau

- La taille (nombre des nœuds) du réseau n'est pas fixe : Les produits peuvent sortir du réseau et des nouveaux produits peuvent entrer dans le réseau.
- Le réseau est en même temps centralisé et décentralisé. Centralisé puisqu'il existe une station de base. Décentralisé, puisque les produits actifs peuvent agir de façon autonome.
- Les nœuds du réseau communiquent entre eux par l'envoi des messages.
- Les nœuds déclenchent des alertes en cas de dépassement des règles de sécurité entre les produits actifs.
- Les nœuds du réseau sont numérotés de façon ascendante.

V.4.5. Caractéristiques du Maître de la Zone

Le réseau est décomposé en quatre zones, et chacune d'elles a un maître qui a les caractéristiques suivantes :

- C'est un capteur en état actif associé à un produit actif.

- Il a l'identifiant le plus grand dans sa zone.
- Il est le responsable de la zone.
- Il est l'intermédiaire entre la station de base et les nœuds appartenant à sa zone.
- Il vérifie le nombre des nœuds redondants dans chaque produit actif, si ce nombre est inférieur ou égal à deux, alors il envoie un message à la station de base pour ajouter des nœuds redondants.
- Il a trois types de tables pour gérer sa zone : la table d'information, la table d'enregistrement et la table grille.

Les tables du maître de la zone :

- **la table d'enregistrement :** Elle contient : les identifiants (IDs) des nœuds de la même zone et leurs positions. Elle permet au maître de la zone de garder la trace des nœuds dans sa zone et aussi de diviser la zone en quarts.

ID du nœud	Position du nœud
1	(20,30)
2	(40,23)
3	(15, 12)

Tableau V-7 Table d'enregistrement

- **la table d'information :**
Elle contient l'ID du nœud envoyant le message, la taille du message et le temps d'arriver du message.

ID du nœud émetteur	Taille du message	Temps d'arrivée du message
1	20	2.001
.	.	.

Tableau V-8 Table d'information

- **la table grille :** Pour la détection des fautes le maître de la zone maintenir **une table grille**, il divise sa zone en une grille et enregistre les informations de sa zone dans cette table grille :

Numéro de carrée	Nœuds de carrée
1	[3 ,12]
2	[7 ,14]

Tableau V-9 Table grille

V.4.6. Caractéristiques du Produit Actif

Le concept de produit dit actif consiste à doter un produit de capacités à communiquer, informer, acquérir, décider et réagir aux stimuli et perturbations de son environnement afin de permettre au produit de s'adapter, d'influer, de coopérer, de transformer le comportement de son environnement (Zouinkhi, 2011). A partir de cette définition, nous déduisons qu'un produit devient actif lorsqu'on lui associe un nœud de capteur. Ce dernier a les capacités de communiquer et d'interagir.

Pour chaque produit du réseau, nous lui associons un nœud principal actif et d'autres nœuds redondants passifs de sorte que tous ces nœuds soient homogènes. Le modèle fonctionnel proposé du produit actif se compose de plusieurs rubriques qui sont les suivants:

a) Un bloc de gestionnaire des dispositifs de perception C'est une interface entre la base de connaissance et les capteurs et les actionneurs.

b) Une base de connaissances Elle transforme les informations perçues par ses propres capteurs et par celles des voisinages en action de sécurité relativement à ses règles décisionnelles qui sont: statique (ambiantes) et communautaire. (Zouinkhi, 2011) La base de connaissances que nous allons utiliser est décrite dans la table V-10:

Identité du produit	Produit (type de produit, symbole, ID de produit, ID du nœud, état du nœud)
Règles statiques	Danger si température (< tempMin ou >tempMax) Bon si température (>tempMin et <tempMax)
Règles de communauté	Distance dépassée si distance entre produits<limite distance.
Actionneurs	Alerte si distance dépassée ou Danger

Tableau V-10 base de connaissance du produit

c) Un bloc de services Dans lequel on trouve les blocs d'inscription qui est lié aux informations de la base de connaissance.

d) Un module de surveillance de voisinage Il collecte les informations des voisinages et les transmet à la base de connaissance pour être traitées.

V.4.7. Fonctionnement de l'Application

Le fonctionnement de l'application se résume en deux phases : celle de l'établissement du réseau et celle du fonctionnement du réseau, et chacune comporte un ensemble de tâches :

V.4.7. 1. Etablissement du réseau

Cette phase comporte les tâches suivantes :

- L'enregistrement de tous les produits actifs existants dans le réseau dans la base de données de la station de base.
- La décomposition du réseau en quatre zones.
- La détermination des maîtres des zones.
- Initialisation des tables des maîtres des zones.

V.4.7.2. Fonctionnement du réseau

Après l'établissement du réseau, il commence à accomplir les tâches suivantes :

- La salutation entre les produits actifs par l'envoi des messages périodiquement.
- La surveillance de l'environnement par les produits qui inter changent toutes les nouveaux évènements dans le réseau.
- La recherche de la panne et sa correction.

V.4.8. Processus de Chaque Fonction dans l'Application

V.4.8.1. Enregistrement des Produits Actifs

Dans cette étape : d'abord on fait l'inscription des PAs dans une table des PAs puis la configuration des produits actifs en leur donnant leurs paramètres de l'identification et des règles de la surveillance. Cette procédure est illustrée dans la figure V-12:

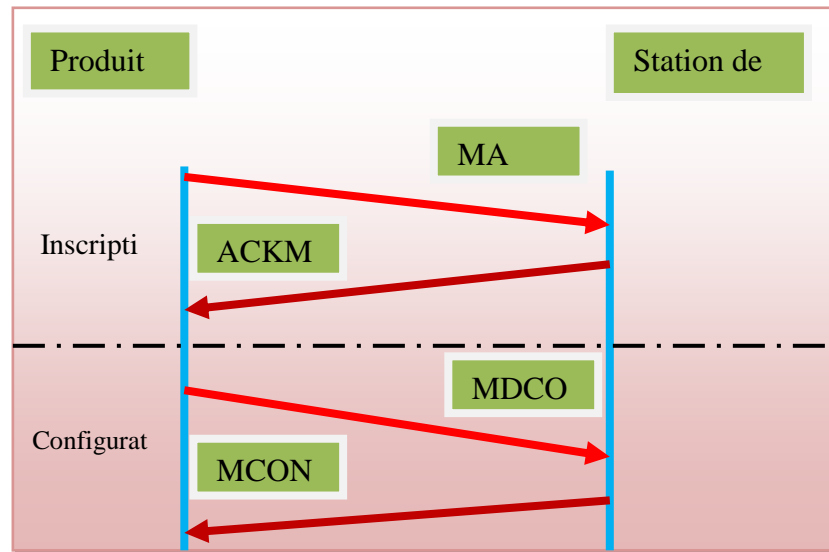


Figure V-12 Scénario d'enregistrement d'un produit actif dans la station de base.

Tel que :

MA : message d'Annonce.

ACKMA : message d'acquittement de **MA**.

MDCONF : message de demande de configuration du produit actif.

MCONF : message contenant la configuration du produit actif.

1) Procédure de numérotation des nœuds des produits actifs :

- ☞ Au début, l'entrepôt ne contient que la station de base, et par la suite les produits actifs entrent dans l'entrepôt un par un.
- ☞ Le produit actif possède un nœud actif et trois nœuds redondants passifs.
- ☞ Le nœud actif sera enregistré dans la station de base (SB) selon le scénario illustré dans la figure V-12.
- ☞ Le numéro du nœud actif du premier produit actif numéroté dans la SB est quatre (4) : Car le nœud actif doit avoir le grand numéro et chaque produit actif a quatre nœuds.
- ☞ Les autres nœuds actifs (NAs) ensuite seront numérotés de façon incrémentée de 4. Par exemple, le numéro du premier nœud actif est 4 et le numéro du deuxième nœud actif est $4+4=8$, et ainsi de suite.

2) La configuration et la numérotation des nœuds redondants :

- Après que le nœud actif (NA) d'un produit actif (PA) est enregistré dans la SB il recopie sa configuration dans les nœuds redondants (NR) selon le pseudo code de la figure V-13:

Pour i de 3 à 1 faire

Eveiller le NR_i .

Copier la configuration dans NR_i .

Mettre le Numéro de NR_i égal à i.

Mettre NR_i dans l'état passif.

Fin

Figure V-13 Pseudo code de configuration des NRs.

- Puis le NA envoie un message à la SB contenant ses NRs avec leurs configuration pour les enregistrer dans sa base des données.

V.4.8.2. Division du Réseau en Quatre Zones

Pour faciliter le contrôle des produits actifs nous allons diviser le réseau en quatre zones principales :

- ∞ Les dimensions de l'entrepôt sont : X_{max} , Y_{max}
- ∞ Pour diviser l'entrepôt nous déterminons d'abord son centre C.
- ∞ Les coordonnées de C sont (X_c , Y_c).
- ∞ les limites de chaque zone sont :
 - ♦ La première zone : $[0, X_c]$ et $[0, Y_c]$.
 - ♦ La deuxième zone : $[0, X_c]$ et $[Y_c, Y_{max}]$.
 - ♦ La troisième zone : $[X_c, X_{max}]$ et $[Y_c, Y_{max}]$.
 - ♦ La quatrième zone : $[X_c, X_{max}]$ et $[0, Y_c]$.
- ∞ Pour déterminer les NAs de chaque zone, on utilise la table des PAs (voir tableau V-11) :

Numéro du produit	ID du NA	Position du NA
1	4	(12 ,5)
2	8	(14,10)

Tableau V-11 Table des produits actifs(PAs)

☞ Le pseudo code pour déterminer les NAs de chaque zone est décrit dans la figure V-14:

Procédure de détermination des nœuds des zones :

TZ1, TZ2,TZ3,TZ4 :tableaux des zones // pour enregistrer les nœuds de chaque zone.

IntZ1, IntZ2, IntZ3, IntZ4 : intervalles des zones // intervalle des coordonnées appartenant à chaque zone.

Pour chaque produit dans la table des PAs **faire**

Si coordonnées de NA appartiennent à IntZ1 **alors**

 Insérer NA dans TZ1.

Sinon

Si coordonnées de NA appartiennent à IntZ2 **alors**

 Insérer NA dans TZ2

Sinon

Si coordonnées de NA appartiennent à IntZ3 **alors**

 Insérer NA dans TZ3

Sinon

Si coordonnées de NA appartiennent à IntZ4 **alors**

 Insérer NA dans TZ4

Fin.

Fin.

Fin.

Fin.

Figure V-14 pseudo code de détermination des nœuds de chaque zone.

V.4.8.3. Détermination des Maîtres des Zones

Nous déterminons le maître de chaque zone par le pseudo code de la figure V-15 :

MZ_k est l'identifiant du maître de la zone k.

TZ_k : tableau // contenant les nœuds de la zone k

IDmax, ID : identifiants (numéros).

Pour k de 1 à 4 **faire**

Initialiser IDmax par zéro.

Pour i de 1 à taille de TZ_k **faire.**

Si ID du NA > IDmax **alors**

Mettre IDmax égal à ID du NA.

Fin.

Fin.

Mettre MZ_k égal à IDmax.

Enregistrer MZ_k dans la base de données. // (voir tableau V-12)

Fin.

Figure V-15 pseudo code de détermination des maîtres des zones

Table des zones et leurs maîtres et leurs nœuds Après la division du réseau en quatre zones et la détermination des maîtres des zones et les nœuds de chaque zone, nous aurons comme résultat une table des maîtres et des nœuds (voir tableau V-12) :

Numéros de zone	1	2	3	4
ID du Maître de zone	40	120	180	192
IDs des nœuds actifs de la zone	4	112	140	188
	8	100	128	148
	12	96		

Tableau V-12 Table des zones avec leurs maîtres et nœuds.

V.4.8.4. Surveillance de L'environnement par les Produits

- ▲ Le produit actif, doit interagir avec son environnement, Ces interactions s'effectuent par le biais des messages suivants :

☞ **Message de salutation** : Ce message est transmis périodiquement entre les produits déjà configurés, il porte les informations propres du produit (identificateur, symboles de sécurité,...), son niveau de sécurité actuel, et a pour rôle ultérieurement de contribuer au processus de calcul de la distance séparant deux produits actifs. Le message de salutation est important pour la surveillance et la communication entre les produits, parce qu'il notifie au produit actif ses caractéristiques principales.

☞ **Message RSI** : L'information de ce message contient principalement la différence de puissance du signal du message de salutation reçu. La valeur de la RSSI est utilisée pour estimer la compatibilité avec la distance minimale entre produits actifs.

- ▲ Après l'étape de configuration, le produit entre dans une boucle définie par deux activités qui sont :
 - la salutation entre les produits par l'envoi du message de salutation.
 - le captage des valeurs ambiantes.
- ▲ Ces deux activités se répètent périodiquement, mais il y a d'autres activités qui se déclenchent de façon événementielle suivant les besoins du superviseur (SB), comme :
 - la demande de lecture des valeurs ambiantes.
 - la demande de lecture des paramètres de configuration.
- ▲ Lorsqu'un PA reçoit un message de salutation, il va :
 - ☞ Envoyer un message RSI précisément vers le produit qui a envoyé le message de salutation en utilisant les informations contenues dans les champs émetteur et récepteur en cas de compatibilité entre PAs.
 - ☞ Ou Déclencher une alarme et émettre un message vers le maître de zone en cas d'incompatibilité et distance dépassée entre PAs. (voir figure V-16 et figure V-17)

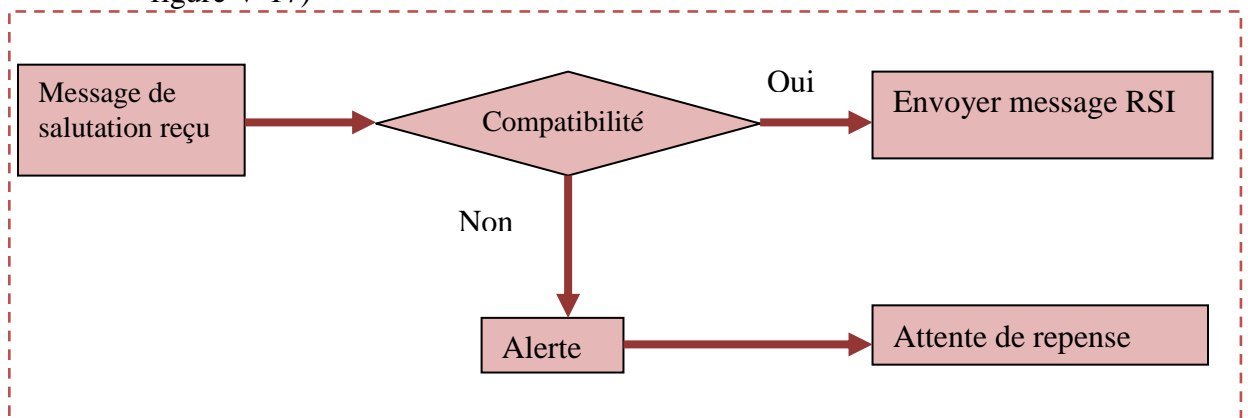


Figure V-16 diagramme de surveillance de compatibilité

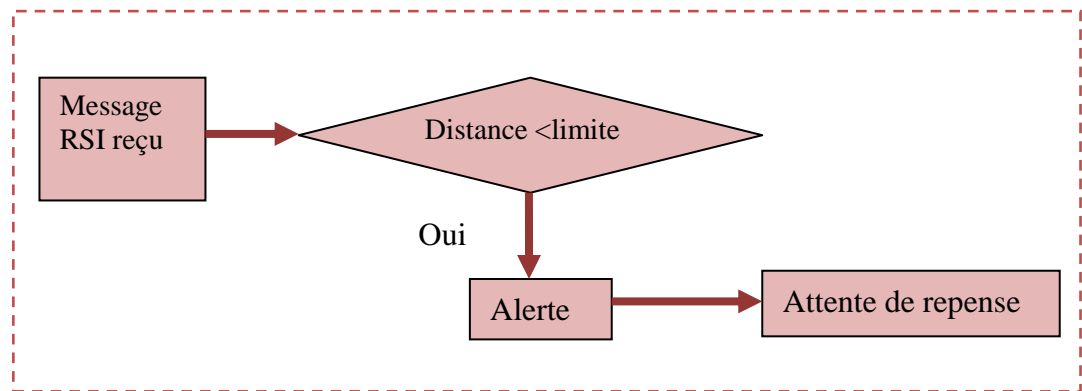


Figure V-17 diagramme de surveillance de la distance.

La surveillance interne du produit actif La surveillance interne consiste à appliquer les règles statiques des valeurs ambiantes sur les mesures captées par le produit actif et puis à déduire l'état du produit actif s'il est en danger ou non ; tel qu'en état de danger, le produit actif déclenche une alerte qui reporte l'état déficient des mesures des valeurs ambiantes. Puis il attend la réponse de la SB.

Si l'état du produit est Bon, alors il forme son message de salutation et l'envoi aux autres produits actifs. (Voir figure V-18)

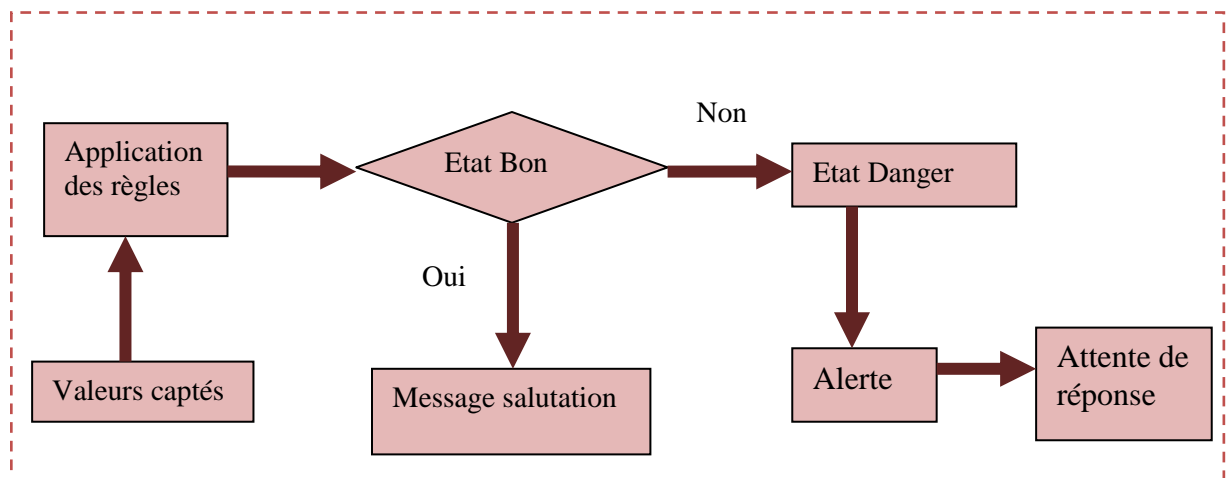


Figure V-18 diagramme de surveillance interne

V.4.8.5. Recherche de la Panne et sa Correction

V.4.8.5.1. Type s des Pannes à Rechercher

Dans notre projet nous nous intéressons à la faute de communication et la faute de sensation.

La détection de la faute de communication se fait par le contrôle du débit de communication du réseau. Le débit de communication se calcule selon l'équation (1).

$$T = \sum_{i=1}^N L / P \quad (1)$$

Tel que :

T : le débit à calculer.

N : nombre des messages reçus par le maître.

L : longueur d'un message.

P : période de temps pendant laquelle le débit est calculé. Une telle période est la différence entre le temps actuel et le temps du dernier débit calculé.

Remarque :

N et **L** sont extraits à partir de la table d'informations du maître de zone.

V.4.8.5.2. Processus de recherche de panne

Après l'établissement du réseau, les maîtres des zones commencent à rechercher l'existence des pannes de communication au niveau des nœuds du réseau, selon le processus suivant :

- 1) Chaque maître de zone surveille sa zone de façon continué.
- 2) Chaque maître de zone calcule le débit de sa zone périodiquement, et le compare à un seuil prédéfinie; si elle est inférieure au seuil, alors il déduit qu'il y a un nœud en panne.
- 3) s'il y a une faute, il va diviser sa zone en quarts virtuels (sous_zone) selon la procédure de division de la figure V-19 :

TZ : la table d'enregistrement du maître de la zone // (voir tableau V-7)

Xmax, Ymax : dimensions de la zone.

(Xc ,Yc) : coordonnées du centre de la zone.

$Xc = Xmax/2$; $Yc=Ymax/2$ // Calcule du centre de la zone

Déterminer les intervalles des quatre sous_zones résultantes.

Pour chaque NA dans TZ **faire**

Comparer la position du NA avec le centre.

Affecter le NA à la sous_zone appropriée.

Fin.

Figure V-19 pseudo code de la procédure de division des zones.

- 4) après la division de la zone le maître remplit sa table grille qui contient les sous_zone (quart) et les nœuds de chaque sous_zone.
- 5) Le maître de zone détermine le quart qui contient la faute : il calcule le débit de chaque quart et le compare à un autre seuil. Si le débit de l'un des quarts est inférieur au seuil, alors il conclut que ce quart contient un nœud en panne.
- 6) Le maître de zone devise ce quart en quatre autres quarts virtuels de la même façon précédente (voir figure V-19).
- 7) Le maître de la zone continue la division en quarts virtuels et la recherche de la faute jusqu'à trouver un quart qui contient un seul nœud actif ou l'absence de panne dans ce quart comme illustré dans la figure V-20:

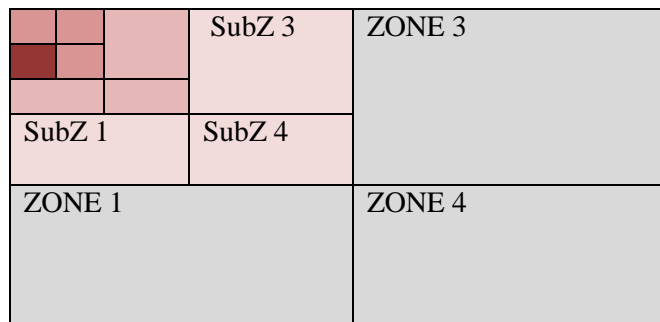


Figure V-20 division successive de zone.

- 8) Le maître de la zone considère que ce nœud est suspect et c'est lui qui rend le débit faible.
- 9) Donc il procède à la vérification de ce nœud.

La recherche du nœud suspect est illustrée dans la figure V-21 .

V.4.8.5.3. Détermination du Nœud en Panne

Lorsque le maître de la zone détecte une panne, il fait une recherche jusqu'à trouver un nœud suspect, puis il procède à la vérification de ce nœud selon les étapes suivantes :

- 1) Le maître de la zone cherche le nœud redondant du nœud suspect dont l'identifiant est ID_1 calculé par l'équation (2) :

$$ID_1 = ID_{NA} - 1 \quad (2)$$
- 2) Après l'identification du NR, il l'éveille pour commencer à sentir.
- 3) Après un temps défini, le maître compare entre les deux nœuds : le nœud suspect et le nœud éveillé selon la procédure de comparaison suivante :

Procédure de comparaison entre deux nœuds : On calcule le débit de chaque nœud. Si la différence entre les valeurs des deux débits est grande par rapport à un seuil défini, et le débit du nœud suspect est inférieur au celle du nœud réveillé, alors on utilise un autre nœud pour déterminer le nœud en panne.

- 4) Si le maître trouve des ambiguïtés pour décider quel est le nœud défectueux, alors un autre nœud redondant sera éveillé, et dont l'identifiant est ID_2 calculé par l'équation (3) :

$$ID_2 = ID_{NA} - 2 \quad (3)$$

- 5) Le troisième nœud va résoudre le conflit du nœud défectueux.
- 6) Le troisième nœud va commencer à sentir.
- 7) Après un temps défini, le maître va faire des comparaisons entre le fonctionnement des trois nœuds : le nœud suspect et les nœuds éveillés.
- 8) Alors le maître peut décider si le nœud suspect souffre d'une faute : Si les débits de communication des deux nœuds éveillés sont similaires et la différence entre ces débits et le débit du nœud suspect est grand alors il décide que le nœud suspect souffre d'une faute.

Substitution du Nœud en Panne par un Nœud Redondant

Après la détermination du nœud en panne, le maître va le remplacer par un nœud redondant, comme suit :

- 1) Le nœud suspect sera mis en état off (désactivé).
- 2) le nœud redondant dont l'identifiant est ID_1 (voir équation 2) restera éveillé et son identifiant sera remplacé par celui du nœud désactivé.
- 3) Alors ce nœud redondant devient le nœud actif.
- 4) et l'autre nœud redondant avec l' ID_2 (voir équation 3) sera mis en état passif.

V.4.8.6. Recherche de la Panne de sensation

En plus de la faute de communication, nous basons aussi sur la faute de sensation pour détecter les nœuds défectueux. La procédure de recherche des nœuds défectueux à base de la faute de sensation est similaire à celle à base de la faute de communication. En résumé le maître de zone fait une comparaison entre les données des nœuds suspects et les données des nœuds redondants réveillés pour décider si un nœud suspect est défectueux. Si c'est le cas il le désactive et active le nœud redondant réveillé.

```

Déterminer le seuil du débit.
Calculer le débit de la zone.
Si (débit < seuil) alors
    Créer TabD. // TabD tableau pour enregistrer les sous zone.
    Créer TabNS // TabNS : tableau pour enregistrer les nœuds suspects.
    Diviser la zone.
    Initialiser compt à 1 // compteur.
    Déterminer le seuil du débit des sous zone SDSZ.
    Tant que compt < taille (TabD) faire
        Calculer le nombre de nœuds dans la sous zone (compt) : NN.
        Calculer le débit de sous zone (compt) : DSZ.
        Si ( (NN>1) et (DSZ<SDSZ) ) alors
            Diviser la sous zone (compt) .
            Mettre à jour TabD // remplacer la sous zone divisé par un tableau des sous
            zone résultantes de la division.
            Mettre Compt =1.
        Sinon
            Si (NN=1) alors
                Mettre le nœud dans TabNS.
                Incrémenter compt par 1.
            Fin si.
        Fin si.
    Fin tant que.
Fin si.
Pour chaque nœud suspect dans TabNS faire
    Eveiller deux nœuds redondants.
    Comparaison(nœud suspect, nœuds redondants).
    Si nœud en panne alors
        Substituer.
    Fin si.
Fin pour.

```

Figure V-21 pseudo code de détection du nœud présentant la faute de communication

V.5. Conclusion

Nous avons présenté dans ce chapitre un aperçu sur la tolérance aux pannes dans les RCSF, les différents types de pannes et la classification des solutions existantes. Puis nous avons décrit deux travaux sur lesquels nous avons basé notre contribution, le premier concerne une application sensible au contexte dont la fonction est le contrôle des produits dans un entrepôt en utilisant un RCSF, le second concerne une méthode de détection des nœuds défectueux dans un RCSF et la substitution de ces nœuds par d'autres nœuds redondants du réseau.

Enfin, nous avons fait une description de notre contribution qui est celle de l'utilisation de la méthode de détection des nœuds défectueux dans l'application sensible au contexte qui se base sur l'idée du produit actif. Cette méthode se base sur la redondance matérielle. Elle a pour rôle de diviser le réseau en quarts et continuer la division des quarts présentant la faute jusqu'à trouver le quart qui contient un seul nœud. Ce dernier sera considéré comme un nœud suspect, et après avoir confirmé qu'il est en panne, il sera remplacé par un autre nœud redondant. Notre contribution est résumée dans la figure V-22.

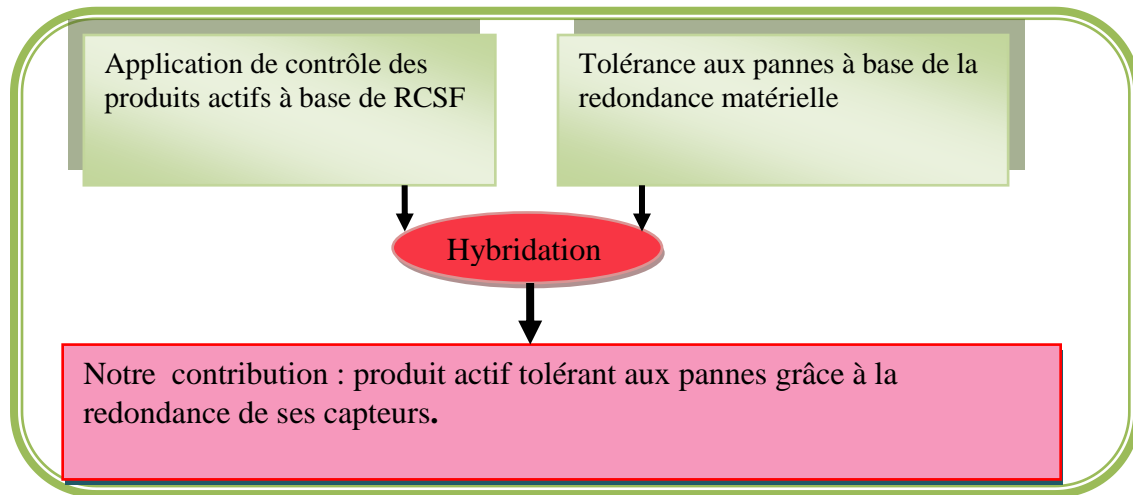


Figure V-22 résumé de notre contribution

Dans le chapitre suivant, nous décrirons l'outil de simulation par lequel nous simulerons notre contribution, et l'implémentation de notre projet.

CHAPITRE VI. IMPLIMENTATION ET SIMULATION

VI.1. Introduction

La simulation des réseaux de capteurs consiste principalement en la reproduction du comportement et du fonctionnement des nœuds capteurs dans un environnement informatique selon la définition suivante : « La simulation consiste à la modélisation informatique d'un système quelconque, en offrant une représentation de toutes les entités de ce système, leurs comportements propres, ainsi que leurs interactions. Elle met à la disposition de l'utilisateur un environnement d'expérimentation dont on peut faire varier les paramètres. » (Imane, 2012)

VI.2. Simulateurs de Réseau Existants

Le point commun entre tous les simulateurs est qu'ils permettent d'évaluer la performance des systèmes avant son déploiement. Il existe plusieurs simulateurs de réseau tel que : NS2, OMNET++, OPNET, GLOMOSIM, JSIM...etc. Dans ce qui suit, nous en décrivons, trois d'entre eux.

1) NS2 C'est un simulateur à événements discrets, écrit en C++ avec une interface TCL. Il est proposé dans le domaine des recherches sur les réseaux. Il est gratuit et open-source. Des améliorations sont en cours sur NS2 avec en parallèle le développement de NS3.

Les nœuds dans le simulateur NS2 modélisent des piles protocolaires OSI complètes qui ne sont pas nécessaires pour représenter des nœuds de capteurs. La simulation des réseaux de capteurs n'est pas facilement supportée par NS2 même si plusieurs travaux sont actuellement en cours pour que NS2 supporte mieux la simulation des réseaux sans fil. (Zouinkhi, 2011)

2) OPNET est un simulateur de réseaux commercialisé par Opnet Technologies, Inc. Une interface utilisateur graphique supporte la configuration des scénarii et le développement des modèles de réseau. Trois niveaux hiérarchiques sont définis pour la configuration.

Le premier, niveau réseau, crée la topologie du réseau à simuler. Le second, niveau nœud, définit le comportement du nœud et contrôle le passage des données entre les différents éléments fonctionnels à l'intérieur du nœud. Enfin, le troisième, niveau processus, décrit les protocoles qui sont représentés par des machines à état fini. Le code source est écrit en C/C++. L'analyse des données simulées est effectuée avec un ensemble de fonctions intégrées. (Zouinkhi, 2011)

3) OMNET++ IDE (Integrated Development Environment) est basé sur la plateforme Eclipse. C'est un environnement open source qui fournit des outils pour la création et la configuration des modèles de réseaux (les fichiers NED et INI) et des outils pour l'exécution d'un lot de programmes ainsi que pour l'analyse des résultats de

simulation. OMNeT++ semble être le meilleur parmi les solutions open source et freeware.

Il séduit de plus en plus la communauté scientifique et un nombre croissant de modèles sont disponibles. (Imane, 2012)

VI.3. Raison de Choisir le Simulateur CASTALIA

Le simulateur, choisi pour nos simulations, est CASTALIA sous la plateforme OMNeT++. Ce dernier est le plus avancé et est le plus utilisé dans la communauté scientifique pour la simulation des réseaux de capteurs sans fil (Imane, 2012). Nous pouvons citer certains de ses avantages :

- Son architecture est modulaire permettant l'intégration de nouveaux modèles.
- Basé sur C++ (récemment, il inclut aussi C#) pour le développement du noyau.
- Les classes de base du simulateur peuvent être étendues et personnalisées.
- Les modèles conçus en l'utilisant sont plus proches de la réalité.

VI.4. Description du Simulateur CASTALIA

Castalia est un simulateur de réseaux de capteurs sans fil (RCSF) et généralement de réseaux de composants à puissance limitée. Il est basé sur la plateforme OMNeT++. Il est utilisé par les chercheurs et les développeurs pour tester des algorithmes distribués et des protocoles avec des composants réels de réseaux de capteurs comme le canal sans fil, le modèle radio, et le comportement des nœuds lié à l'accès au canal radio.

Castalia a été développé à NICTA (National ICTAustralia) en 2006. En 2007, il est devenu publique en tant que projet open source sous la licence publique académique. (Zouinkhi, 2011)

VI.4.1. Structure du code source de CASTALIA

La structure du code source de CASTALIA est hiérarchique. Chaque module est un répertoire qui contient des sous répertoires. Dans le cas d'un module simple, le répertoire contient du code C++ pour décrire le comportement du module, un fichier *.ned pour définir la structure du module et un fichier *.msg pour spécifier les paquets que génèrent ce module.

Un nœud dans un réseau de capteurs sous CASTALIA est un module composé essentiellement d'un canal sans fil, de la couche physique radio et de la couche MAC. Les développeurs de Castalia ont implémenté toutes les caractéristiques et les détails des composants d'un réseau de capteurs dans leur simulateur. (Zouinkhi, 2011)

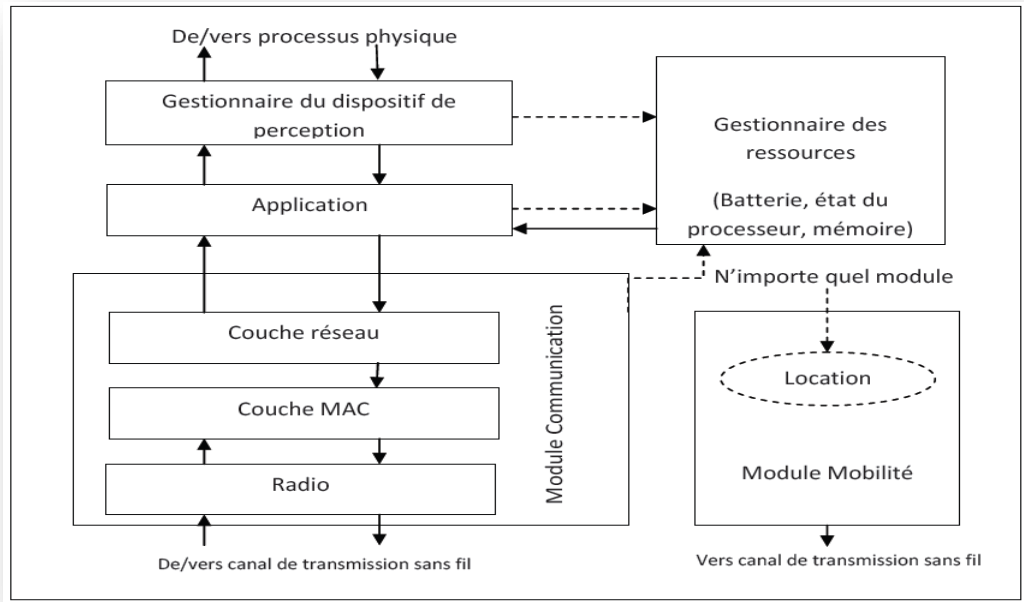


Figure VI-1 Structure d'un nœud sous Castalia.

1) Module radio Ce module supporte plusieurs états : en transmission, en réception/attente, en veille. Il est possible de faire varier les puissances et les délais de transmission pour les différents états.

2) Module MAC Le module MAC (Medium Access Control) est une partie importante dans le comportement du nœud. La première motivation dans le développement de Castalia a été de tester le réglage du protocole MAC dans des conditions réalistes du canal radio.

Les paramètres les plus importants de cette couche sont :

- Cycle d'utilisation : c'est la fraction du temps que le nœud passe dans l'écoute du canal. La valeur $(1 - \text{cycle d'utilisation})$ est la fraction de temps de veille. Le réglage de ce paramètre est important pour minimiser la consommation d'énergie.
- Intervalle d'écoute : c'est la période d'écoute du canal pour un nœud.

Pour limiter le risque, on utilise le protocole CSMA/CA : Le protocole CSMA/CA utilise un mécanisme d'esquive de collision basé sur un principe d'accusé de réception réciproque entre l'émetteur et le récepteur.

3) Module réseau À partir de la version 1.2, le simulateur introduit le module réseau (routage). Au début, ils n'ont pas donné une importance à cet élément, donc il n'y avait pas de module pour le routage. L'utilisateur doit alors traiter le routage des paquets dans le module application s'il en a besoin.

4) Module Processus physique Pour avoir un environnement de simulation proche de la réalité, on a besoin d'avoir des modèles flexibles de processus physique (corrélation spatiale des données, variabilité dans le temps, ...). Pour ce besoin, le

simulateur offre un module générique pour le processus physique pour fournir les données aux nœuds de capteurs.

La base de ce modèle est les valeurs des sources dont l'influence est diffusée dans l'espace. Les sources peuvent changer leurs positions et leurs valeurs. L'effet des sources multiples dans un point est additif.

5) Module du gestionnaire du dispositif de perception C'est un module intermédiaire entre le module application et le module processus physique (mesure captée). L'application envoie les requêtes de lecture des valeurs ambiantes vers ce module, et ce dernier accède au processus physique pour lire la valeur. Ce module permet de simuler la réception de données (événements).

Il faut savoir que dans le cas pratique, les valeurs proposées par le dispositif physique sont faussées par l'inexactitude du dispositif de perception. Castalia propose un ensemble de paramètres pour modéliser cette inexactitude. Dans Castalia, il y a une seule correspondance entre le processus physique et le dispositif de perception. En pratique, un processus physique unique peut déclencher plusieurs dispositifs de perception.

6) Module application C'est le module dans lequel, l'utilisateur effectue normalement beaucoup de changements pour implémenter de nouveaux algorithmes. Dans ce module, nous avons défini les messages du modèle du produit actif de notre projet.

7) Module mobilité Le module de mobilité spécifie comment les nœuds bougent dans l'espace. Les autres modules peuvent y accéder à n'importe quel moment. Des notifications périodiques sont envoyées au canal sans fil pour donner la position des nœuds.

VI.5. Simulation

VI.5.1. Environnement de Simulation

Les caractéristiques du PC sur lequel les simulations ont été effectuées sont :

- Processeur : Intel(R) Core(TM) 2 Duo CPU T5800 @ 2.00 GHz 2.00 GHz
- RAM : 2.00 Go
- système d'exploitation : Ubuntu 14.04 LTS
- plateforme de développement : OMNET++
- simulateur : CASTALIA

VI.5.2. Paramètres Utilisés dans la Simulation

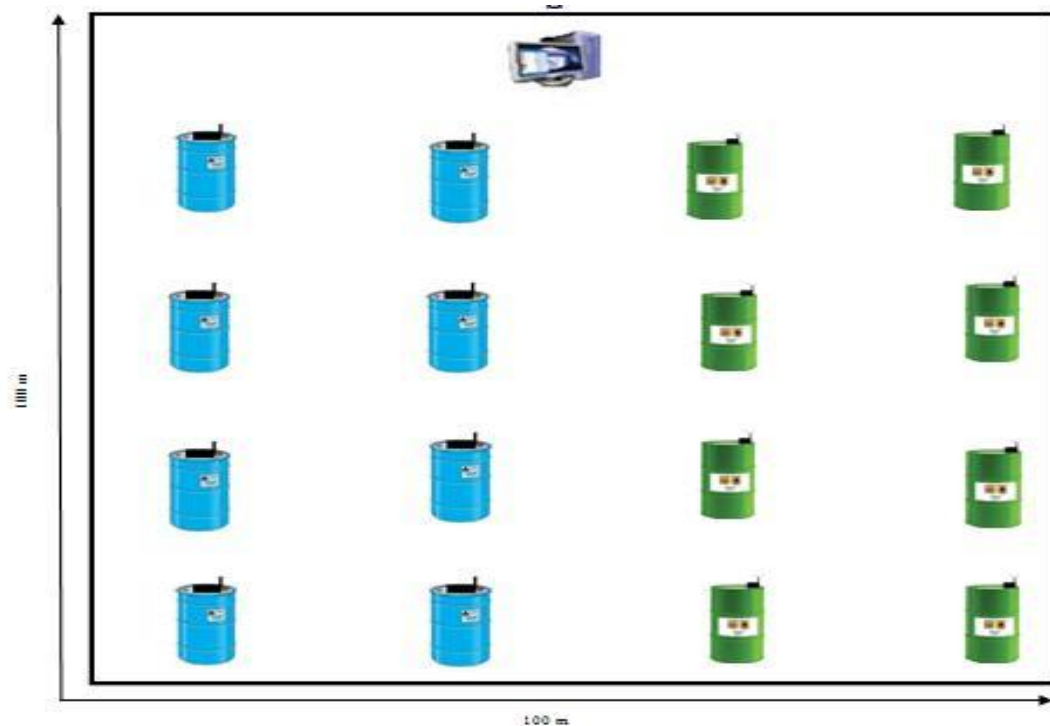


Figure VI-2 Disposition des produits actifs dans un entrepôt 100m x 100m

La surface simulée est un entrepôt avec une surface de 100m x 100m qui contient 16 produits actifs (Chaque conteneur dispose d'un capteur actif et 3 capteurs passifs). Comme illustré dans la figure VI-2.

Nous avons un réseau de capteurs sans fil qui comporte 65 capteurs considérant l'un des capteurs représente la station de base. Nous avons deux types de produits : produits toxiques et produits inflammables.

Les produits sont organisés suivant une matrice de 4 produits actifs en ligne x 4 produits actifs en colonne, et la distance qui sépare deux produits voisins sur un même axe est de 5m.

VI.6. Résultats de Simulation

VI.6.1. Salutation entre les Produits Actifs

Dans ce processus, chaque produit actif envoie à ses voisins des messages de salutations contenant ses propres informations, comme le montre la figure VI-3.

```

Castalia-Trace.txt x
1
2 0.027539895218 SN.node[0].Application This is the base station: nothing to do!
3 5.103570918737 SN.node[1].Application Sending [Context Hello Packet] of size 25 bytes to communication Layer
4 5.103570918737 SN.node[1].Application Node[1] a diffusé un paquet Hello
5 5.103570918737 SN.node[1].Application Packet size = 50, transmission will last 0.0016 secs
6 5.105190918736 SN.node[0].Application Noeud[0] a reçu un paquet Hello à partir du noeud[1]
7 5.105190918736 SN.node[2].Application Noeud[2] a reçu un paquet Hello à partir du noeud[1]
8 5.105190918736 SN.node[3].Application Noeud[3] a reçu un paquet Hello à partir du noeud[1]
9 5.105190918736 SN.node[4].Application Noeud[4] a reçu un paquet Hello à partir du noeud[1]
10 5.105190918736 SN.node[5].Application Noeud[5] a reçu un paquet Hello à partir du noeud[1]
11 5.105190918736 SN.node[6].Application Noeud[6] a reçu un paquet Hello à partir du noeud[1]
12 5.105190918736 SN.node[7].Application Noeud[7] a reçu un paquet Hello à partir du noeud[1]
13 5.105190918736 SN.node[8].Application Noeud[8] a reçu un paquet Hello à partir du noeud[1]
14 5.105190918736 SN.node[9].Application Noeud[9] a reçu un paquet Hello à partir du noeud[1]
15 5.105190918736 SN.node[10].Application Noeud[10] a reçu un paquet Hello à partir du noeud[1]
16 5.105190918736 SN.node[11].Application Noeud[11] a reçu un paquet Hello à partir du noeud[1]
17 5.105190918736 SN.node[12].Application Noeud[12] a reçu un paquet Hello à partir du noeud[1]
18 5.105190918736 SN.node[13].Application Noeud[13] a reçu un paquet Hello à partir du noeud[1]
19 5.105190918736 SN.node[14].Application Noeud[14] a reçu un paquet Hello à partir du noeud[1]
20 5.105190918736 SN.node[15].Application Noeud[15] a reçu un paquet Hello à partir du noeud[1]
21 5.105190918736 SN.node[16].Application Noeud[16] a reçu un paquet Hello à partir du noeud[1]
22 5.105190918736 SN.node[17].Application Noeud[17] a reçu un paquet Hello à partir du noeud[1]
23 5.105190918736 SN.node[18].Application Noeud[18] a reçu un paquet Hello à partir du noeud[1]
24 5.105190918736 SN.node[19].Application Noeud[19] a reçu un paquet Hello à partir du noeud[1]
25 5.105190918736 SN.node[20].Application Noeud[20] a reçu un paquet Hello à partir du noeud[1]
26 5.105190918736 SN.node[21].Application Noeud[21] a reçu un paquet Hello à partir du noeud[1]
27 5.105190918736 SN.node[22].Application Noeud[22] a reçu un paquet Hello à partir du noeud[1]
28 5.105190918736 SN.node[23].Application Noeud[23] a reçu un paquet Hello à partir du noeud[1]
29 5.105190918736 SN.node[24].Application Noeud[24] a reçu un paquet Hello à partir du noeud[1]
30 5.105190918736 SN.node[25].Application Noeud[25] a reçu un paquet Hello à partir du noeud[1]
31 5.105190918736 SN.node[26].Application Noeud[26] a reçu un paquet Hello à partir du noeud[1]
32 5.105190918736 SN.node[27].Application Noeud[27] a reçu un paquet Hello à partir du noeud[1]
33 5.105190918736 SN.node[28].Application Noeud[28] a reçu un paquet Hello à partir du noeud[1]
34 5.105190918736 SN.node[29].Application Noeud[29] a reçu un paquet Hello à partir du noeud[1]
35 5.105190918736 SN.node[30].Application Noeud[30] a reçu un paquet Hello à partir du noeud[1]
36 5.105190918736 SN.node[31].Application Noeud[31] a reçu un paquet Hello à partir du noeud[1]
37 5.105190918736 SN.node[32].Application Noeud[32] a reçu un paquet Hello à partir du noeud[1]
38 5.105190918736 SN.node[33].Application Noeud[33] a reçu un paquet Hello à partir du noeud[1]

```

Figure VI-3 Salutation entre les produits actifs

VI.6.2. Vérification de la Compatibilité

Dans ce processus, chaque produit vérifie sa compatibilité avec ses voisins

A. Cas des produits compatibles Dans ce cas, deux produits voisins de même type sont compatibles, comme illustré dans la figure VI-4.

```

a-Trace.txt (-/Castalia-3.3/Simulations/dataFusion) - gedit
Ouvrir Enregistrer Annuler
Castalia-Trace.txt x
4289 11.446351425178 SN.node[62].Application Noeud[62] a reçu un paquet Hello à partir du noeud[64]
4290 11.446351425178 SN.node[63].Application Noeud[63] a reçu un paquet Hello à partir du noeud[64]
4291 11.502285113521 SN.node[13].Application Le noeud[13] vérifie la compatibilité de son produit avec ceux de ces voisins.
4292 11.502285113521 SN.node[13].Application Le produit du noeud[13] est Toxique
4293 11.502285113521 SN.node[13].Application Le produit de son voisin; noeud[1] est Toxique
4294 11.502285113521 SN.node[13].Application Produits compatibles! Rten à signaler.
4295 11.502285113521 SN.node[13].Application Le produit de son voisin; noeud[5] est Toxique
4296 11.502285113521 SN.node[13].Application Produits compatibles! Rten à signaler.
4297 11.502285113521 SN.node[13].Application Le produit de son voisin; noeud[9] est Toxique
4298 11.502285113521 SN.node[13].Application Produits compatibles! Rten à signaler.
4299 11.502285113521 SN.node[13].Application Le produit de son voisin; noeud[17] est Toxique
4300 11.502285113521 SN.node[13].Application Produits compatibles! Rten à signaler.
4301 11.502285113521 SN.node[13].Application Le produit de son voisin; noeud[21] est Toxique
4302 11.502285113521 SN.node[13].Application Produits compatibles! Rten à signaler.

```

Figure VI-4 Cas des produits compatibles

B. Cas des produits incompatibles

Dans ce cas, deux produits voisins de types différents sont incompatibles, comme illustré dans la figure VI-5.

```

Castalia-Trace.txt (-/Castalia-3.3/Simulations/dataFusion) - gedit
Castalia-Trace.txt x
4289 11.446351425178 SN.node[62].Application Noeud[62] a reçu un paquet Hello à partir du noeud[64]
4290 11.446351425178 SN.node[63].Application Noeud[63] a reçu un paquet Hello à partir du noeud[64]
4291 11.502285113521 SN.node[13].Application Le noeud[13] vérifie la compatibilité de son produit avec ceux de ces
voisins.
4292 11.502285113521 SN.node[13].Application Le produit du noeud[13] est Toxique
4293 11.502285113521 SN.node[13].Application Le produit de son voisin; noeud[1] est Toxique
4294 11.502285113521 SN.node[13].Application Produits compatibles! Rien à signaler.
4295 11.502285113521 SN.node[13].Application Le produit de son voisin; noeud[5] est Toxique
4296 11.502285113521 SN.node[13].Application Produits compatibles! Rien à signaler.
4297 11.502285113521 SN.node[13].Application Le produit de son voisin; noeud[9] est Toxique
4298 11.502285113521 SN.node[13].Application Produits compatibles! Rien à signaler.
4299 11.502285113521 SN.node[13].Application Le produit de son voisin; noeud[17] est Toxique
4300 11.502285113521 SN.node[13].Application Produits compatibles! Rien à signaler.
4301 11.502285113521 SN.node[13].Application Le produit de son voisin; noeud[21] est Toxique
4302 11.502285113521 SN.node[13].Application Produits compatibles! Rien à signaler.
4303 11.502285113521 SN.node[13].Application Le produit de son voisin; noeud[25] est inflammable
4304 11.502285113521 SN.node[13].Application Produits incompatibles!
4305 11.502285113521 SN.node[13].Application La distance minimale, entre les produits incompatibles, doit être égale

```

Figure VI-5 Cas des produits incompatibles

VI.6.3. Cas du Déclenchement d'Alerte

La figure VI-6 montre le cas du déclenchement d'alerte qui correspond au cas du dépassement de la distance entre deux produits incompatibles. Dans l'exemple illustré, nous avons mis la distance entre les deux produits 21 et 25 égale à 2m et ces produits sont incompatibles car le produit 21 est toxique et le produit 25 est inflammable, donc après la vérification de la distance entre ces deux produits le produit 21 a envoyé un paquet d'alerte au maître de sa zone puisque la distance calculée (2m) est inférieure à la distance de sécurité (5m).

```

4527 11.507597444117 SN.node[21].Application Le produit de son voisin; noeud[21] est Toxique
4528 11.507597444117 SN.node[21].Application Produits compatibles! Rien à signaler.
4529 11.507597444117 SN.node[21].Application Le produit de son voisin; noeud[25] est inflammable
4530 11.507597444117 SN.node[21].Application Produits incompatibles!
4531 11.507597444117 SN.node[21].Application La distance minimale, entre les produits incompatibles, doit être égale
à 5m
4532 11.507597444117 SN.node[21].Application Le noeud[21] vérifie sa distance avec son voisin; noeud[25].
4533 11.507597444117 SN.node[21].Application Distance = 2(m)
4534 11.507597444117 SN.node[21].Application Distance insuffisante!
4535 11.507597444117 SN.node[21].Application Noeud[21] a envoyé un paquet d'alerte à son cluster-head; noeud[5]
4536 11.507597444117 SN.node[21].Application Sending [Context Alert Packet] of size 25 bytes to communication layer
4537 11.507597444117 SN.node[21].Application Le produit de son voisin; noeud[29] est inflammable
4538 11.507597444117 SN.node[21].Application Produits incompatibles!
4539 11.507597444117 SN.node[21].Application La distance minimale, entre les produits incompatibles, doit être égale
à 5m
4540 11.507597444117 SN.node[21].Application Le noeud[21] vérifie sa distance avec son voisin; noeud[29].
4541 11.507597444117 SN.node[21].Application Distance courante = 10(m)
4542 11.507597444117 SN.node[21].Application Distance suffisante! Aucune alerte à envoyer

```

Figure VI-6 cas du déclenchement d'alerte

La figure VI-7 montre le cas du déclenchement d'une fausse alerte en cas de nœud défectueux, le produit 61 est défectueux, il a déclenché une alerte malgré l'absence des conditions de déclenchement d'alerte : la distance réelle entre le produit 61 et son voisin le produit 25 est égale à 12.8m, mais après vérification de la distance, le produit 61 a montré que cette distance est égale à 0.9m, donc il a envoyé une alerte.


```

Castalia-Trace.txt (-/Castalia-3.3/Simulations/dataFusion) - gedit
Castalia-Trace.txt x
4484 11.506683931241 SN.node[61].Application Le noeud[61] vérifie sa distance avec son voisin; noeud[25].
4485 11.506683931241 SN.node[61].Application Distance réelle = 12.8062(m)
4486 11.506683931241 SN.node[61].Application Distance suffisante! Normalement, rien à signaler.
4487 11.506683931241 SN.node[61].Application Mais le noeud[61] est défectueux.
4488 11.506683931241 SN.node[61].Application Distance mesurée = 0.900218m
4489 11.506683931241 SN.node[61].Application Le noeud[61] va envoyer un résultats erroné (fausse alerte)
4490 11.506683931241 SN.node[61].Application Noeud[61] a envoyé un paquet d'alerte à son cluster-head; noeud[41]
4491 11.506683931241 SN.node[61].Application Sending [Context Alert Packet] of size 25 bytes to communication layer
4492 11.506683931241 SN.node[61].Application Le produit de son voisin; noeud[29] est inflammable
4493 11.506683931241 SN.node[61].Application Produits incompatibles!
4494 11.506683931241 SN.node[61].Application La distance minimale, entre les produits incompatibles, doit être égale
à 5m

```

Figure VI-7 cas du déclenchement de fausse alerte

VI.6.4. Détection du Nœud Défectueux

La figure VI-8 montre une partie de la trace de notre application concernant la détection du nœud défectueux par le maître de la zone (cluster-head), et la confirmation de sa défaillance en éveillant l'un de ses nœuds redondants.

Dans cet exemple, le produit 41 est un maître de zone, il a reçu une alerte à partir du produit 61, il vérifie si l'alerte est juste en éveillant l'un des nœuds redondants du produit 61, qui est le nœud 62, puis le nœud 62 a calculé la distance entre le produit 25 et le produit 61. La distance trouvée est de 12.8m.

```

4561 11.507597444117 SN.node[21].Application Produits compatibles! Rien à signaler.
4562 11.50830393124 SN.node[41].Application Noeud[41] a reçu un paquet d'alerte à partir du noeud[61]
4563 11.50830393124 SN.node[41].Application Le noeud[41] vérifie si le noeud, qui a envoyé l'alerte, fonctionne
normalement,
4564 11.50830393124 SN.node[41].Application en envoyant un paquet wake-up à l'un des noeuds redondants de ce noeud.
4565 11.50830393124 SN.node[41].Application Le noeud[41] cherche le noeud redondant dans la table des voisins...
4566 11.50830393124 SN.node[41].Application Noeud redondant trouvé: noeud[62]
4567 11.50830393124 SN.node[41].Application Le noeud le plus proche du noeud[61] est le noeud[62]
4568 11.50830393124 SN.node[41].Application Sending [Context Wakeup Packet] of size 25 bytes to communication layer
4569 11.50830393124 SN.node[41].Application Le noeud[41] a envoyé un paquet Wake-up au noeud redondant; noeud[62]
4570 11.510290843349 SN.node[62].Application Le noeud [62] a reçu un paquet wake-up à partir du noeud[41]
4571 11.510290843349 SN.node[62].Application ce noeud se charge de la vérification de la compatibilité des produits.
4572 11.510290843349 SN.node[62].Application Le noeud[62] (redondant) vérifie la compatibilité de son produit avec
celui du noeud[25]
4573 11.510290843349 SN.node[62].Application Le produit du noeud[62] est Toxique
4574 11.510290843349 SN.node[62].Application Le produit de son voisin; noeud[25] est inflammable
4575 11.510290843349 SN.node[62].Application Produits incompatibles!
4576 11.510290843349 SN.node[62].Application La distance minimale, entre les produits incompatibles, doit être égale
à 5m
4577 11.510290843349 SN.node[62].Application Le noeud[62] vérifie sa distance avec son voisin; noeud[25].
4578 11.510290843349 SN.node[62].Application Distance courante = 12.8062(m)
4579 11.510290843349 SN.node[62].Application Distance suffisante! Il faut signaler que l'ancienne alerte était
fausse.
4580 11.510290843349 SN.node[62].Application Noeud[62] a envoyé un paquet wake up response à son cluster-head; noeud
[41]
4581 11.510290843349 SN.node[62].Application Sending [Context Wake up Response Packet] of size 25 bytes to
communication layer

```

Figure VI-8 détection du nœud défectueux

VI.6.5. Confirmation de la panne

Après la détection du nœud défectueux, le maître confirme que ce nœud est défectueux en comparant les données du nœud défectueux et celles de son nœud redondant qu'on a éveillé. Dans la figure VI-9, la distance calculée par le nœud 61 est 0.9m et la distance calculée par le nœud redondant 62 est 12.8m qui est la distance réelle. Donc le maître a déduit que l'alerte est fausse et que le nœud 61 est défectueux.

```

la-Trace.txt (~-/Castalia-3.3/Simulations/dataFusion) - gedit
Castalia-Trace.txt x
4798 11.514774417127 SN.node[41].Application Le produit de son voisin; noeud[53] est Toxique
4799 11.514774417127 SN.node[41].Application Produits compatibles! Rien à signaler.
4800 11.514774417127 SN.node[41].Application Le produit de son voisin; noeud[57] est inflammable
4801 11.514774417127 SN.node[41].Application Produits incompatibles!
4802 11.514774417127 SN.node[41].Application La distance minimale, entre les produits incompatibles, doit être égale
à 5m
4803 11.514774417127 SN.node[41].Application Le noeud[41] vérifie sa distance avec son voisin; noeud[57].
4804 11.514774417127 SN.node[41].Application Distance courante = 5(m)
4805 11.514774417127 SN.node[41].Application Distance suffisante! Aucune alerte à envoyer
4806 11.514774417127 SN.node[41].Application Le produit de son voisin; noeud[61] est Toxique
4807 11.514774417127 SN.node[41].Application Produits compatibles! Rien à signaler.
4808 11.521153842336 SN.node[41].Application Noeud[41] a reçu un paquet wake up response à partir du noeud[62]
4809 11.521153842336 SN.node[41].Application Le noeud[41] vérifie si l'alerte du noeud[61] est fausse ou non
4810 11.521153842336 SN.node[41].Application L'alerte du noeud[61] est fausse
4811 11.521153842336 SN.node[41].Application Distance entre noeud[25] et noeud[62] = 12.8062
4812 11.521153842336 SN.node[41].Application Distance entre noeud[25] et noeud[61] = 0.900218
4813 11.521153842336 SN.node[41].Application Le noeud[61] est défectueux
4814 11.521153842336 SN.node[41].Application Le noeud[41] envoie un paquet de désactivation à ce noeud
4815 11.521153842336 SN.node[41].Application Sending [Context Deactivation Packet] of size 25 bytes to communication
laver

```

Figure VI-9 confirmation de la défaillance

VI.6.6. Correction de la panne

Après la détection du nœud défectueux, et la confirmation de sa défaillance, on le désactive, et en même temps on active le nœud redondant qu'on a éveillé, et ce nœud devient le nœud actif pour le produit actif. La figure VI-10 montre un exemple de ce cas qui correspond à la désactivation du nœud défectueux 61. Donc le nœud redondant 62 qu'on a éveillé sera le nœud actif qui remplace le nœud désactivé 61.

```

Castalia-Trace.txt (~-/Castalia-3.3/Simulations/dataFusion) - gedit
Castalia-Trace.txt x
5102 11.538258554595 SN.node[33].Application Le produit de son voisin; noeud[45] est Toxique
5103 11.538258554595 SN.node[33].Application Produits compatibles! Rien à signaler.
5104 11.538258554595 SN.node[33].Application Le produit de son voisin; noeud[49] est Toxique
5105 11.538258554595 SN.node[33].Application Produits compatibles! Rien à signaler.
5106 11.538258554595 SN.node[33].Application Le produit de son voisin; noeud[53] est Toxique
5107 11.538258554595 SN.node[33].Application Produits compatibles! Rien à signaler.
5108 11.538258554595 SN.node[33].Application Le produit de son voisin; noeud[57] est inflammable
5109 11.538258554595 SN.node[33].Application Produits incompatibles!
5110 11.538258554595 SN.node[33].Application La distance minimale, entre les produits incompatibles, doit être égale
à 5m
5111 11.538258554595 SN.node[33].Application Le noeud[33] vérifie sa distance avec son voisin; noeud[57].
5112 11.538258554595 SN.node[33].Application Distance courante = 11.1803(m)
5113 11.538258554595 SN.node[33].Application Distance suffisante! Aucune alerte à envoyer
5114 11.538258554595 SN.node[33].Application Le produit de son voisin; noeud[61] est Toxique
5115 11.538258554595 SN.node[33].Application Produits compatibles! Rien à signaler.
5116 11.538543087329 SN.node[61].Application Le noeud [61] a reçu un paquet de désactivation à partir du noeud[41]
5117 11.538543087329 SN.node[61].Application ce noeud ne se charge plus maintenant de la surveillance de la zone
5118 11.538543087329 SN.node[61].Application Le noeud [61] éteint la radio.

```

Figure VI-10 Désactivation du nœud défectueux

VI.7. Conclusion

Dans ce chapitre nous avons abordé une description du simulateur CASTALIA, qui est spécifique à la simulation des réseaux de capteurs sans fil. Parmi ses caractéristiques importantes, nous citons la capacité de donner des résultats plus proches de la réalité. C'est pour cette raison que nous avons choisi ce simulateur.

Nous avons réalisé notre projet sous le simulateur CASTALIA ; grâce à cette simulation nous avons découvert pratiquement ce simulateur qui nous permet de simuler la sensibilité du contexte par les réseaux de capteurs sans fil de façon très proche de la réalité.

Nous avons prouvé par cette simulation la faisabilité de notre contribution qui s'établit en deux parties :

- 1) une application sensible au contexte à base du RCSF.
- 2) détection des nœuds défectueux et la correction par le remplacement du nœud défectueux par un nœud redondant.

Nous signalons que ce simulateur (CASTALIA) est récent (développé en 2007), nous l'avons appris personnellement en basant sur son manuel(en Anglais) puisque il y a peu de gens qui l'utilisent.

Nous avons obtenu des résultats encourageants et faisables dans la réalité.

CONCLUSION GENERALE

L'informatique pervasive assure la présence de l'informatique partout à tout moment et pour tous. Dans ce domaine, on trouve les systèmes sensibles au contexte. Ces systèmes utilisent les capteurs pour collecter les informations à partir de leur environnement ; grâce aux caractéristiques des réseaux de capteurs sans fil(RCSF). L'utilisation de ces réseaux dans les systèmes sensibles au contexte fournit un fonctionnement permanent et performant de ces systèmes.

Aujourd'hui les systèmes sensibles au contexte sont très liés au RCSF. Parmi les caractéristiques importantes des RCSF c'est la tolérance aux pannes qui permet la continuité du fonctionnement de façon correcte.

Notre recherche se focalise sur deux concepts : les systèmes sensibles au contexte et les réseaux de capteurs sans fil. Nous avons montré la relation entre eux à partir d'une étude des travaux qui relient ces deux concepts.

Notre contribution consiste à utiliser une méthode de tolérance aux pannes dans les RCSF pour améliorer le fonctionnement d'un système sensible au contexte. Pour cela, nous avons étendu le travail de ZOUINKHI (ZOUINKHI, 2011), qui est un système sensible au contexte à base du RCSF. Il concerne le contrôle des produits chimiques dans un entrepôt, tel que chaque produit est un produit actif qui interagit avec son contexte grâce aux capteurs lui y sont associés.

Notre contribution était d'associer à chaque produit un ensemble de capteurs au lieu d'un seul ; un seul capteur est actif alors que les autres (redondants) sont passifs. Puis, faire la correction de panne par la méthode décrite dans (Taleb, 2009) qui consiste à détecter exactement les nœuds défectueux et les remplacer par des nœuds redondants. Notre contribution complète le travail de (Trab Sourour, 2013). Qui consiste à détecter les fausses alertes générées dans le système de (ZOUINKHI, 2011), sans recourir à la correction.

Nous avons prouvé par la simulation la faisabilité de notre contribution qui est constituée de deux parties :1) une application sensible au contexte à base du RCSF. 2) détection des nœuds défectueux et la correction par le remplacement du nœud défectueux par un nœud redondant. Nous avons obtenu des résultats encourageants et faisables dans la réalité.

Nous signalons que le simulateur (CASTALIA) est récent (développé en 2007), et il est recommandé pour la simulation de la sensibilité au contexte par les RCSF.

REFERENCES

- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002).** *A survey on sensor networks. Communications magazine, IEEE, 40(8), 102-114.*
- Ali, Y. (2007).** *Modélisation et composition de services sensibles au contexte*(Doctoral dissertation, Université Abderrahmane Mira de Béjaia).
- Amin, M. (2011).** *Conception d'une architecture journalisée tolérante aux fautes pour un processeur à pile de données* (Doctoral dissertation, Metz).
- Amina, B. O. U. D. J. A. A. D. A. R. (2010).** Plateforme Basée Agents Pour l'Aide à la Conception et la Simulation des Réseaux de Capteurs Sans Fil.
- Baker, N., Zafar, M., Moltchanov, B., & Knappmeyer, M. (2009, April).** Context-Aware Systems and Implications for Future Internet. In *Future Internet Assembly* (pp. 335-344).
- Beydoun, K. (2009).** *Conception d'un Protocole de Routage Hierarchique pour les Reseaux de Capteurs* (Doctoral dissertation, PhD. Thesis, L'UFR des Sciences et Techniques de l'Universite de Franche-Comte, Franche-Comte).
- Brézillon, P., Borges, M., Pino, J. A., & Pomerol, J. C. (2004, July).** Context-awareness in group work: Three case studies. In *Proc. of*.
- Brown, P. J. (1995).** The stick-e document: a framework for creating context-aware applications. *ELECTRONIC PUBLISHING-CHICHESTER-*, 8, 259-272.
- Brown, P. J., Bovey, J. D., & Chen, X. (1997).** Context-aware applications: from the laboratory to the marketplace. *Personal Communications, IEEE, 4(5), 58-64.*
- Chaari, T. (2007).** *Adaptation d'applications pervasives dans des environnements multi-contextes* (Doctoral dissertation, institut national des sciences appliquées de Lyon).
- Challal, Y. (2008).** Réseaux de Capteurs Sans Fils. *Cours, Systèmes Intelligents pour le Transport, Université de Technologie de Compiègne, France, 17.*
- Chalmers, D. (2011).** *Sensing and systems in pervasive computing: Engineering context aware systems.* Springer.
- Chen, G., & Kotz, D. (2000).** *A survey of context-aware mobile computing research* (Vol. 1, No. 2.1, pp. 2-1). Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.
- Chong, S. K., Krishnaswamy, S., & Loke, S. W. (2005, March).** A context-aware approach to conserving energy in wireless sensor networks. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on* (pp. 401-405). IEEE.
- Conchon, E. (2007).** *Définition et mise en oeuvre d'une solution d'émulation de réseaux sans fil* (Doctoral dissertation, Institut National Polytechnique De Toulouse).
- Culler, D., Estrin, D., & Srivastava, M. (2004).** *Guest editors' introduction: overview of sensor networks. Computer, 37(8), 41-49.*
- De Sousa, G. (2008).** *Etude en vue de la réalisation de logiciels bas niveau dédiés aux réseaux de capteurs sans fil: microsystème de fichiers* (Doctoral dissertation, Université Blaise Pascal-Clermont-Ferrand II).
- Deng, Z., & Zhu, M. (2010, August).** A context aware motor controller based on WSN. In *Mechatronics and Automation (ICMA), 2010 International Conference on* (pp.

- 1501-1506). IEEE.
- Deng, Z., & Zhu, M. (2010, August).** A context aware motor controller based on WSN. In *Mechatronics and Automation (ICMA), 2010 International Conference on* (pp. 1501-1506). IEEE.
- Dey, A. K. (2001).** *Understanding and using context. Personal and ubiquitous computing, 5(1), 4-7.*
- Dey, A. K., Abowd, G. D., & Salber, D. (2001).** *A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-computer interaction, 16(2), 97-166.*
- ELGhayam, Y. (2011).** *La Sensibilité au Contexte dans un Environnement Mobile* (Doctoral dissertation, Université Mohammed V Souissi – RABAT).
- Elnahrawy, E., & Nath, B. (2004).** Context-aware sensors. In *Wireless Sensor Networks* (pp. 77-93). Springer Berlin Heidelberg.
- Girolami, M., Lenzi, S., Furfari, F., & Chessa, S. (2008, September).** Sail: A sensor abstraction and integration layer for context awareness. In *Software Engineering and Advanced Applications, 2008. SEAA'08. 34th Euromicro Conference* (pp. 374-381). IEEE.
- Henricksen, K., Indulsk, J., & Rakotonirainy, A. (2002).** Modeling context information in pervasive computing systems. In *Pervasive Computing* (pp. 167-180). Springer Berlin Heidelberg.
- Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., Altmann, J., & Retschitzegger, W. (2003, January).** Context-awareness on mobile devices-the hydrogen approach. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on* (pp. 10-pp). IEEE.
- Huairfeng, Q., & Xingshe, Z. (2005, November).** Context aware sensor network. In *Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing* (pp. 1-7). ACM.
- Hwang, J., & Yoe, H. (2011).** Study on the context-aware middleware for ubiquitous greenhouses using wireless sensor networks. *Sensors, 11(5), 4539-4561.*
- Imad, B. (2011).** *Analyse des données d'un système RFID en vue de sa sûreté de fonctionnement* (Doctoral dissertation, Université d'Oran-Sénia).
- IMANE, Niar Leila.** *Analyse graphique pour la surveillance dans un RCSF.* Diss. Université des sciences et de la technologie Mohamed Boudiaf d'Oran.
- Ismail, B. (2010).** *Une architecture fondée sur les services pour la prise en compte du contexte dans les systèmes d'information ubiquitaires* (Doctoral dissertation, Université Badji Mokhtar de Annaba).
- Karine El Makssoud Épouse ABBAS.** *Système d'Accès Personnalisé à l'Information: Application au Domaine Médical.* Thèse de doctorat, INSA de Lyon, 2008
- Karl, H., & Willig, A. (2003).** A short survey of wireless sensor networks. *Telecommunication Networks group, Technical Report.*
- Khelladi, L., & Badache, N. (2004).** Les réseaux de capteurs: état de l'art. *Faculté électronique et informatique Bab Ezzouar-Algérie.*
- Krumm, J. (Ed.). (2009).** *Ubiquitous computing fundamentals.* CRC Press.
- Lamine, M. M. (2007).** *Sécurité dans les Réseaux de Capteurs Sans-Fil. Mémoire de Magistère en Informatique Ecole Doctorale d'Informatique de Bejaia, 2008.*
- Loke, S. (2006).** *Context-aware pervasive systems: architectures for a new breed of*

- applications*. CRC Press.
- Makhoul, A. (2008).** *Réseaux de capteurs: localisation, couverture et fusion de données (Doctoral dissertation)*.
- Marzencki, M., Hung, B., Lin, P., Huang, Y., Cho, T., Chuo, Y., & Kaminska, B. (2010, April).** Context-aware physiological data acquisition and processing with wireless sensor networks. In *Medical Measurements and Applications Proceedings (MeMeA), 2010 IEEE International Workshop on* (pp. 53-56). IEEE.
- Miraoui, M. (2009).** *Architecture logicielle pour l'informatique diffuse: modélisation du contexte et adaptation dynamique des services (Doctoral dissertation, École de technologie supérieure)*.
- NAILI, M.** *Gestion de contexte dans l'habitat communicant: localisation indoor et fourniture de services sensibles au contexte par ontologies (Doctoral dissertation, Université Abderrahmane Mira de Béjaia)*.
- Park, I., Choi, H., Shin, K. Y., Kim, J., & Park, S. (2006, February).** CAS/sup 2: context awareness service system based on wireless sensor networks. In *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference* (Vol. 2, pp. 4-pp). IEEE.
- Pascoe, J. (1998, October).** Adding generic contextual capabilities to wearable computers. In *Wearable Computers, 1998. Digest of Papers. Second International Symposium on* (pp. 92-99). IEEE.
- Petrelli, D., Not, E., Strapparava, C., Stock, O., & Zancanaro, M. (2000, April).** Modeling context is like taking pictures. In *Proc. of the Workshop "The What, Who, Where, When, Why and How of Context-Awareness" in CHI2000*.
- Popovici, D. (2012).** *Gestion du contexte pour des applications mobiles dédiées aux transports (Doctoral dissertation, Université de Valenciennes et du Hainaut-Cambresis)*.
- Poslad, S. (2009).** *Front Matter* (pp. i-xxvii). John Wiley & Sons, Ltd.
- Raicu, I., Richter, O., Schwiebert, L., & Zeadally, S. (2004).** Using Wireless Sensor Networks to Narrow the Gap between Low-Level Information and Context-Awareness. *arXiv preprint cs/0411039*.
- Reetz, E. S., Tonjes, R., & Baker, N. (2010, May).** Towards global smart spaces: Merge wireless sensor networks into context-aware systems. In *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on* (pp. 337-342). IEEE.
- Ryan, N. S., Pascoe, J., & Morse, D. R. (1998).** Enhanced reality fieldwork: the context-aware archaeological assistant. In *Computer applications in archaeology*. Tempus Reparatum.
- Sashima, A., Inoue, Y., Ikeda, T., Yamashita, T., & Kurumatani, K. (2008, December).** CONSORTS-S: A mobile sensing platform for context-aware services. In *Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008. International Conference on* (pp. 417-422). IEEE.
- Schilit, B., Adams, N., & Want, R. (1994, December).** Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on* (pp. 85-90). IEEE.
- Schilit, B. N., & Theimer, M. M. (1994).** Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5), 22-32.

- Soukkarieh, B. (2010).** *Technique de l'internet et ses langages: vers un système d'information web restituant des services web sensibles au contexte*(Doctoral dissertation, Université de Toulouse, Université Toulouse III-Paul Sabatier).
- Tahayori, H., Pagani, E., Degli Antoni, G., & Astaneh, M. S. (2007, February).** Enhanced Sensor Network: A Specialized Infrastructure For Context-Aware Applications. In *Iadis International Conference Applied Computing* (pp. 568-572). Nuno Guimarães and Pedro Isafas.
- Taleb, A. A., Pradhan, D. K., & Kocak, T. (2009, June).** A Technique to Identify and Substitute Faulty Nodes in Wireless Sensor Networks. In *Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on* (pp. 346-351). IEEE.
- Trab, S., Zouinkhi, A., Boussaid, B., & Abdelkrim, M. N. (2013, May).** Application of distributed fault detection in WSN to dangerous chemical products based on Bayesian approach. In *Advanced Logistics and Transport (ICALT), 2013 International Conference on* (pp. 129-134). IEEE.
- Ward, A., Jones, A., & Hopper, A. (1997).** A new location technique for the active office. *Personal Communications, IEEE*, 4(5), 42-47.
- Willig, H. K. A., & Karl, H. (2005).** Protocols and architectures for wireless sensor networks. *England: John Wiley & Sons*.
- Winograd, T. (2001).** Architectures for context. *Human-Computer Interaction*,16(2), 401-419.
- XUE, Y., AGUILAR, A., GONZALEZ, A., & BARROUX, M. (2010).** Agrégation de données dans les réseaux de capteurs. *rapport final*.
- Zouinkhi, A. (2011).** *Contribution à la modélisation de produit actif communicant, Spécification et Evaluation d'un protocole de communication orienté sécurité des produits* (Doctoral dissertation, Université Henri Poincaré-Nancy I).