UNIVERSITÉ DE LORRAINE

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

THESE

Présentée pour l'obtention du titre de

Docteur de l'Université de Lorraine

en Automatique, traitement du signal et des images, génie informatique

Et

Docteur de l'École Nationale d'Ingénieurs de Gabès

Spécialité Génie Electrique

par

Kais MEKKI

Gestion de l'information embarquée dans des matériaux communicants à l'aide de protocoles de réseaux de capteurs sans fil

---

Soutenue publiquement le 02/06/2016 devant le jury composé de :

RAPPORTEURS :	MR. THIERRY VAL	PROFESSEUR, UNIVERSITE TOULOUSE
	MR. ABDELAZIZ BOURAS	PROFESSEUR, UNIVERSITE LYON 2
EXAMINATEURS :	MR. MOHAMED AOUN	PROFESSEUR, UNIVERSITE GABES
	MR. ANDRE THOMAS	PROFESSEUR, UNIVERSITE LORRAINE (CO-ENCADRANT)
	MR. ERIC RONDEAU	PROFESSEUR, UNIVERSITE LORRAINE (DIRECTEUR)
	MR. MOHAMED NACEUR ABDELKRIM	PROFESSEUR, UNIVERSITE GABES (DIRECTEUR)
	MR. WILLIAM DERIGENT	MAITRE DE CONFERENCES, UNIVERSITE LORRAINE (CO-DIRECTEUR)

Centre de Recherche en Automatique de Nancy, UMR 7039 Université de Lorraine - CNRS

Table des matières

Introduction générale 1
Chapitre I : Le concept de matière communicante 5
I. Introduction 6
II. Du produit intelligent à la matière communicante 6
II.1. Produits Intelligents : Etat de l'art 6
II.2. Bénéfices générés par les produits intelligents 8
II.3. Limites de concept de produit intelligent 8
II.4. Le nouveau paradigme de matière communicante 9
II.5. Premiers travaux menés dans le contexte de matière communicante 10
II.5.1. Auto-mesure des produits intelligents 10
II.5.2. Auto-mesure SMS dans un nanoréseau de capteurs sans fil 11
II.5.3. Textile communicant 12
II.5.4. Traçabilité du bois par marquage chimique 12
III. Gestion de cycle de vie d'un « système matière communicante » 13
III.1. Le concept PLM 13
III.2. Architecture PLM de « système matière communicante » 14
III.3. Processus de dissémination d'informations dans un système matière communicante 15
III.3.1. Identification et évaluation de la pertinence des « informations produit » 16
a) Identification des « informations produit » au sein de la base de données 16
b) Evaluation de la pertinence des « informations produit » 17
III.3.2. Stockage des données pertinentes dans la matière communicante 17
IV. Problématique de thèse 18
IV.1. Contexte 18
IV.2. Premières applications de produits intelligents dans la construction béton 19
IV.2.1. La maturométrie 19
IV.2.2. La détection de fissures 19
IV.2.3. L'identification du béton 20
IV.2.4. Marquage CE d'une poutre en béton 20
IV.3. Intérêts de la matière communicante dans la construction béton 20
IV.4. Positionnement 22
IV.4.1. Positionnement technique 22
IV.4.2. Positionnement scientifique 23
V. Conclusion 26

Chapitre II : Etat de l'art sur les protocoles de stockage et de lecture dans les réseaux de capteurs..... 27
I. Introduction..... 28
II. Les réseaux de capteurs sans fil..... 28
II.1. Architecture des réseaux de capteurs..... 28
II.2. Caractéristiques des réseaux de capteurs..... 29
II.3. Contraintes des réseaux de capteurs..... 30
II.4. Architecture d'un nœud de capteurs..... 30
II.5. Modèles de communication..... 31
II.5.1. La couche application..... 32
II.5.2. La couche transport..... 32
II.5.3. La couche réseau..... 32
II.5.4. La couche liaison de données..... 32
II.5.5. La couche physique..... 32
II.5.6. Plan de gestion d'énergie..... 32
II.5.7. Plan de gestion de mobilité..... 32
II.5.8. Plan de gestion de tâche..... 32
II.6. Les modèles de transmission des données dans les réseaux de capteurs..... 33
II.6.1. Le modèle orienté événement..... 33
II.6.2. Le modèle orienté requête..... 33
II.6.3. Le modèle à flux continu..... 33
III. Protocoles de stockage d'information dans les réseaux de capteurs..... 33
III.1. Protocoles de dissémination réactifs..... 34
a) ALURP..... 34
b) QBDCS..... 34
c) MobiQuery..... 35
d) BPPDD..... 36
III.2. Protocoles de dissémination proactifs..... 36
III.2.1. Protocoles structurés..... 36
a) TTDD..... 36
b) HexDD..... 37
c) EAPDD..... 37
d) RailRoad..... 37
e) LBDD..... 38
f) QDD..... 38
III.2.2. Protocoles non structurés..... 39

a) DEEP 39
b) RaWMS 39
c) DBDD 40
d) XYLS 40
e) Supple 41
f) TwinRoute 41
III.3. Analyse et synthèse 42
IV. Protocoles de lecture et recherche d'information dans les réseaux de capteurs 45
IV.1. Protocoles à deux phases requête/réponse 45
a) Directed Diffusion 46
b) EAR 47
c) GBR 47
d) BMR 47
IV.2. Protocoles à une seule phase requête/réponse 48
IV.2.1. Protocoles basés sur le coût de liaison 48
a) MCFA 48
b) GRAB 48
c) MARP 49
IV.2.2. Protocoles multi-chemins 49
a) REAMR 49
b) EQSR 50
c) MR2 50
d) N-to-1 51
IV.2.3. Protocoles basés sur un agent de recherche 51
a) Rumor Routing 51
b) EEQR 52
IV.3. Analyse et synthèse 53
V. Conclusion 55
Chapitre III : Spécification de protocoles de stockage et de lecture d'informations dans la matière 56
I. Introduction 57
II. Notion de noeud maitre 57
III. Stockage des informations dans la matière communicante 58
III.1. Inondation dans les réseaux de capteurs sans fil 59
III.1.1. Techniques d'inondation 59
a) Distance-based 59

b) Location-based 60
c) Cluster-based 60
d) Probabilistic-based 60
e) Counter-based 60
f) Analyse et critères de choix 60
III.1.2. Mécanisme de diffusion « counter-based » 61
III.2. Stratégies de stockage dans les réseaux de capteurs sans fil 62
III.3. Le protocole de dissémination USEE 63
III.3.1. Etats des nœuds de capteurs dans le processus de dissémination 63
III.3.2. Modèlisation par Réseau de Petri d'un noeud dans le processus de dissémination 64
a) Mode de réception 64
b) Mode de stockage 65
c) Mode de diffusion 66
III.3.3. Entête de message 67
III.3.4. Algorithme de dissémination 68
IV. Lecture d'information dans la matière communicante 70
IV.1. Random Walk et Pull Gossip 71
IV.2. Algorithme de sélection des nœuds de bordure du voisinage 73
IV.2.1. Techniques d'estimation de distance dans les réseaux de capteurs 73
a) Techniques basées sur la propagation du signal radio 73
b) Techniques basées sur la cardinalité d'intersection de voisinage 75
c) Analyse et critères de choix 76
IV.2.2. Algorithme de sélection des nœuds voisins distants 77
a) Entêtes des messages d'algorithme de sélection des nœuds voisins distants 78
b) Fonctionnement de l'algorithme de sélection des nœuds voisins distants 79
IV.3. Protocole de recherche d'information RaWPG 80
IV.3.1. Fonctionnement du protocole RaWPG 80
IV.3.2. Fonction de coût de liaison 82
IV.3.3. Entêtes des messages de RaWPG 82
IV.3.4. Algorithme de lecture RaWPG 84
a) Traitement du message REQUEST 84
b) Traitement du message REQUEST_NG 85
c) Traitement du message RESPONSE ou FAILURE 86
V. Conclusion 87

Chapitre IV : Evaluation de protocoles de stockage et de lecture d'informations dans la matière 88
I. Introduction 89
II. Les simulateurs de réseaux de capteurs sans fil 89
II.1. NS3 89
II.2. COOJA 89
II.3. Prowler 90
II.4. TOSSIM 90
II.5. SENSE 90
II.6. Castalia 90
III. Le simulateur Castalia 91
III.1. Le module canal sans fil 91
III.2. Le module radio 92
III.3. Le module MAC 92
III.4. Le module réseau 92
III.5. Le module processus physique 92
III.6. Le module de gestion du dispositif de perception 92
III.7. Le module de gestion des ressources 92
III.8. Le module application 92
III.9. Le module mobilité 93
IV. Etude des informations du cycle de vie du préfabriqué à considérer 93
IV.1. BOL (Beginning Of Life – début de vie) 94
IV.2. MOL (Middle Of Life – usage) 95
IV.3. EOL (End Of Life – fin de vie) 96
IV.4. Synthèse 96
V. Environnement et paramètres de simulation 97
VI. Instrumentation d'un réseau de capteurs dans un préfabriqué en béton 98
VI.1. Choix des micro-nœuds de capteurs 98
VI.2. Validation de la propagation du signal radio 2.4 GHZ dans la matière béton 100
VI.3. Etude de la densité des nœuds de capteurs dans un préfabriqué 101
VI.3.1. Etude de la densité des nœuds en fonction de l'uniformité de stockage probabiliste d'USEE ... 101
a) Etude théorique 101
b) Etude par simulation 103
VI.3.2. Etude de la densité des nœuds en fonction de l'efficacité de sélection des nœuds de bordure .. 104
VII. Evaluation de performances du protocole de dissémination USEE 105
VII.1. Plan d'évaluation des performances 105

VII.2. Critères d'évaluation des performances 105
VII.3. Capacité de stockage de la matière 106
VII.4. USEE vs. Protocoles de littérature 108
VII.4.1. Uniformité de réplication 109
VII.4.2. Taux de réplication 110
VII.4.3. Délai de dissémination 111
VII.4.4. Nombre de communications 112
VII.4.5. Consommation d'énergie 113
VIII. Evaluation des performances de protocole de lecture RaWPG 114
VIII.1. Plan d'évaluation des performances 114
VIII.2. Critères d'évaluation des performances 115
VIII.3. Etude de la longueur maximale de chemin de recherche TTL 115
VIII.4. RaWPG vs. Simple Random Walk + Pull Gossip 119
VIII.4.1. Taux de couverture 119
VIII.4.2. Taux de succès 120
VIII.4.3. Délai de lecture 120
VIII.5. RaWPG vs. MARP 121
VIII.5.1. Taux de succès 122
VIII.5.2. Taux de duplication 122
VIII.5.3. Nombre de communications 123
VIII.5.4. Consommation d'énergie 124
IX. Synthèse et discussion 124
Conclusion générale 132
Annexe 136
Liste des publications 151
Bibliographie 152

Liste des figures

Figure 1. Transformation du bâtiment en objet communicant 2
Figure 2. SMS: Self-Measurement System 11
Figure 3. Auto-mesure SMS dans un nanoréseau de capteurs sans fil 11
Figure 4. Processus de dissémination de donnée dans un textile communicant.....12
Figure 5. Vue générale du système PLM et ses objectifs 14
Figure 6. Système d'information à deux niveaux lié au « système matière communicante ».....15
Figure 7. Vue d'une partie du MLD de la table MaterialDefinition et des « tuples produit » 16
Figure 8. Table MaterialDefinition et pertinence des data items issus des « tuples produit » 17
Figure 9. Production d'un textile communicant 18
Figure 10. Lecture de l'identifiant des tags RFID par un opérateur.....20
Figure 11. Dispersion des micro-nœuds dans un préfabriqué en béton.....23
Figure 12. Première architecture pour la gestion des informations dans le béton 24
Figure 13. Uniformité de stockage des informations dans le réseau de capteurs du préfabriqué.....25
Figure 14. Architecture d'un réseau de capteurs sans fil.....29
Figure 15. Types de communication dans un réseau de capteurs sans fil 29
Figure 16. Architecture d'un nœud de capteurs sans fil 31
Figure 17. Pile protocolaire des réseaux de capteurs sans fil (Akyildiz et al., 2002).....31
Figure 18. Les protocoles de stockage de données dans les réseaux de capteurs sans fil 34
Figure 19. Dissémination réactive par MobiQuery 35
Figure 20. Stratégies de stockage proactif structuré dans les réseaux de capteurs sans fil 36
Figure 21. Partitionnement de réseau en quatre quadrants logiques successifs 39
Figure 22. Réplication d'information par marche aléatoire 40
Figure 23. Les protocoles de recherche d'information dans les réseaux de capteurs sans fil 45
Figure 24. Directed Diffusion 46
Figure 25. Phase de renforcement du protocole BMR 48
Figure 26. Construction des chemins entre sources et puits dans N-to-1 51
Figure 27. Connexion de l'utilisateur à la matière par le nœud maitre 57
Figure 28. Diagramme de séquence d'interaction utilisateur/matériel pour le stockage et la lecture.....58
Figure 29. Structuration du protocole de dissémination USEE.....59
Figure 30. Variation des états des nœuds pour la dissémination des informations par USEE 63
Figure 31. Modèle RdP d'un noeud dans un processus de dissémination.....64
Figure 32. Modèle RdP du processus de dissémination de données dans la matière communicante.....65

Figure 33. Modèle RdP du mode réception (p2) 65
Figure 34. Conversion de l'importance d'information en probabilité de stockage P 66
Figure 35. Modèle RdP du mode stockage (p3) 66
Figure 36. Modèle RdP du mode diffusion (p4).....67
Figure 37. Modèle RdP complet d'un nœud dans le processus de dissémination USEE 67
Figure 38. Structure du message DATA-MSG 68
Figure 39. Effet du stockage contrôlé dans un voisinage 68
Figure 40. Fonctionnement de Random Walk et Random Walk+Pull Gossip 71
Figure 41. Ensemble des nœuds de bordure dans un voisinage de nœud de capteurs.....73
Figure 42. Effet du choix des noeuds voisins distants.....73
Figure 43. Techniques d'estimation de distance : ToA et TDoA 74
Figure 44. Intersection des cercles de communication radio de deux nœuds voisins n1 et n2.....75
Figure 45. Variation de cardinalité de la zone d'intersection M suivant la distance entre n1 et n2 78
Figure 46. Structure du message HELLO 78
Figure 47. Structure du message TAB_EXCHANGE.....78
Figure 48. Structure du message FAR_NEIGHBORS 78
Figure 49. Recherche d'information initiée par le nœud maitre.....80
Figure 50. Recherche d'information au niveau de noeud Y 81
Figure 51. Acheminement de requêtes et réponses dans le réseau de capteurs par RaWPG.....82
Figure 52. Structure du message REQUEST 82
Figure 53. Structure du message REQUEST_NG.....83
Figure 54. Structure du message FAR_INFO 83
Figure 55. Structure du message RESPONSE 84
Figure 56. Structure du message FAILURE.....84
Figure 57. Structure d'un nœud sous Castalia.....91
Figure 58. Cycle de vie du préfabriqué 93
Figure 59. Structure hiérarchique du SensorCube.....99
Figure 60. Validation de la propagation du signal radio 2.4 GHZ dans le béton 101
Figure 61. Variation théorique de probabilité d'existence d'information 103
Figure 62. Taux d'existence de l'information dans les voisinages.....104
Figure 63. Résultats de sélection des nœuds voisins distants (bordures) 104
Figure 64. Consommation d'énergie des modules d'un nœud de capteurs 106
Figure 65. Résultats de réplication d'USEE avec NS vs. USEE sans NS 107
Figure 66. Taux de réplication d'USEE avec NS vs. USEE sans NS 108
Figure 67. Capacité de stockage de la matière d'USEE avec NS vs. USEE sans NS 108

Figure 68. Division de la matière en 100 cellules de dimension 109
Figure 69. Uniformité de la réplication d'USEE vs. Protocoles de littérature 110
Figure 70. Taux de réplication d'USEE vs. Protocoles de littérature.....111
Figure 71. Délai de dissémination d'USEE vs. Protocoles de littérature 112
Figure 72. Nombre de communications d'USEE vs. Protocoles de littérature 113
Figure 73. Consommation d'énergie moyenne d'USEE vs. Protocoles de littérature.....114
Figure 74. Recherche par diffusion multi-sauts dans le réseau enfoui dans la matière.....116
Figure 75. Modélisation du schéma de Bernoulli par arbre de recherche d'une information 117
Figure 76. Longueur maximale de TTL suivant la probabilité d'existence d'information 118
Figure 77. Taux de couverture de RaWPG vs. Simple Random Walk+ Pull Gossip.....120
Figure 78. Taux de succès de RaWPG vs. Simple Random Walk+ Pull Gossip 120
Figure 79. Délai de lecture de RaWPG vs. Simple Random Walk+ Pull Gossip.....121
Figure 80. Fonctionnement du protocole de recherche d'information MARP.....122
Figure 81. Taux de succès de RaWPG vs. MARP 122
Figure 82. Taux de duplication de RaWPG vs. MARP.....123
Figure 83. Nombre de communications de RaWPG vs. MARP 123
Figure 84. Consommation d'énergie moyenne de RaWPG vs. MARP.....124
Figure 85. Taux de consommation d'énergie pour les données internes et externes 129
Figure 86. Evolution de la consommation d'énergie des nœuds durant le cycle de vie.....131

Liste des tableaux

Tableau 1. Résumé des protocoles de stockage d'information dans les réseaux de capteurs sans fil 43
Tableau 2. Résumé des protocoles de recherche d'information dans les réseaux de capteurs sans fil...54
Tableau 3. Techniques d'inondation dans les réseaux de capteurs sans fil 59
Tableau 4. Techniques de stockage dans les réseaux de capteurs sans fil.....62
Tableau 5. Informations de la phase BOL.....94
Tableau 6. Informations de la phase MOL.....96
Tableau 7. Informations de la phase EOL.....97
Tableau 8. Tableau récapitulatif de l'étude et des contraintes idoines sur l'instrumentation.....97
Tableau 9. Paramètres de simulation.....98
Tableau 10. Résumé des micro-nœuds de capteurs sans fil 100
Tableau 11. Nombre de transmissions du nœud maitre.....113
Tableau 12. Durée de vie de nœud vs. Capacité de batterie.....128
Tableau 13. Consommation d'énergie des données internes et externes.....129

Glossaire

ADV	ADVertise
AHP	Analytic Hierarchy Process
ALURP	Adaptive Local Update-based Routing Protocol
AoA	Angle of Arrival
AP	Aggregation Point
APR	Alternative Path Reinforcement
BAN	Body Area Networks
BMR	Braided Multipath Routing
BOL	Beginning Of Life
BPPDD	Balancing Push and Pull Data Dissemination
CH	Cluster Head
CM	Cluster Members
DBDD	Double-Blind Data Discovery
DCHT	Delay-Constrained High-Throughput Protocol for Multipath Transmission
DD	Directed Diffusion
DEEP	Density-based proactivE data dissEmination Protocol
DRR	Directional Rumor Routing
EAPDD	Edge Agregation Point for Data Dissemination
EAR	Energy Aware Routing
EEQR	Energy Efficient Query-based Routing protocol
EOL	End Of Life
EQSR	Energy efficient and QoS based multi-path Routing
ETX	Expected Transmission Count
GBR	Gradient Based Routing
GPS	Global Positioning System
GRAB	GRAdient Broadcast
HexDD	Hexagonal cell-based Data Dissemination
IEEQR	Improved Energy Efficient Query-based Routing
INFREQ	INFormation REQuest
IoT	Internet of Things

LBDD	Line-Based Data Dissemination
LiFi	Light Fidelity
MAC	Medium Access Control
MCFA	Minimum Cost Forwarding Algorithm
MLD	Modèle Logique de Données
MLE	Maximum-Likelihood Estimator
MOL	Middle Of Life
MR2	Maximally Radio-Disjoint Multipath Routing
NIDES	Neighborhood Intersection Distance Estimation Scheme
NS	Node Storage
N-to-1	N-to-1 Multipath Routing Protocol
PG	Pull Gossip
PLM	Product Lifecycle Management
PPR	Primary Path Reinforcement
QBDCS	Query-Based Data Collection Scheme
QDD	Quadtree-based Data Dissemination
RAD	Random Assessment Delay
RaWMS	RAndom Walk based Membership Service
RaWPG	Random Walk and Pull Gossip
RdP	Réseau de Petri
REAMR	Reliable and Energy-Aware Multipath Routing
RFID	Radio Frequency IDentification
RLW	Random Line Walk
RQN	Résonance Quadripolaire Nucléaire
RR	Rumor Routing
RSS	Received Signal Strength
RW	Random Walk
SMS	Self-Measurement System
TDoA	Time Difference of Arrival
ToA	Time of Arrival
ToS	Type of Service
TTDD	Two-Tier Data Dissemination
TTL	Time To Live
USEE	Uniform Storage and Energy Efficient
WSN	Wireless Sensor Networks

Introduction générale

Cette thèse aborde le problème de la dissémination des informations liées au produit tout au long de son cycle de vie, par l'exploitation du concept de matière communicante. Ce concept de matière communicante représente une évolution récente du produit intelligent et est une instanciation originale de l'usage de l'Internet des Objets.

Depuis la fin du 20ème siècle, les recherches consistant à rendre intelligents des produits manufacturiers ou de consommation passent par la mise en place d'un système électronique sur le produit lui-même pour lui assurer des fonctions de mémorisation, de communication et de traitements (pour les systèmes les plus évolués). Les technologies utilisées peuvent être simplement des puces RFID qui le plus souvent ne permettent que d'associer un pointeur pour accéder à des informations déportées dans des serveurs mais encore peuvent être des nœuds de capteurs avec des communications sans fil. Ces capteurs permettent alors d'ajouter une fonction supplémentaire de contrôle de l'environnement du produit ou du produit lui-même (McFarlane et al., 2012). Ce type de produit a cependant des limites (Kubler, 2012) : i) le flux d'informations relatif au produit est rompu si la puce RFID ou le nœud de capteurs est endommagé, ii) la transformation du produit peut générer des trous dans le système d'information (par exemple lorsque le produit est transformé/découpé, certaines pièces résultantes seront vierges d'informations), iii) les technologies RFID et nœuds de capteurs actuels ont des limitations (capacité mémoire, débit, énergie, etc.).

Pour résoudre ces limites, le concept de matière communicante a été proposé par (Thomas, 2009) au Centre de Recherche en Automatique de Nancy. C'est un nouveau paradigme qui change radicalement la manière de voir le produit et la matière. Ce concept consiste à donner la faculté au produit d'être intrinsèquement et intégralement communicant. Intrinsèquement signifie que le flux matière et le flux informationnel sont complètement interdépendants et fusionnés au sein d'une même entité. De façon idéale, on pourrait étendre ce concept au niveau moléculaire, même si une telle approche est encore loin d'être réaliste au regard des limitations technologiques actuelles. Certaines recherches semblent néanmoins prometteuses, comme les travaux de (Peña et al., 2011), (Kubler, 2012), (Krommenacker et al., 2013), (Jover, 2013) et (Tran-Dang et al., 2014). Ces derniers travaux ont été proposés principalement pour des applications d'auto-mesure et de traçabilité du textile et bois tout au long de leur cycle de vie. Les travaux de cette thèse ont quant à eux comme contexte l'application du concept de la matière communicante dans l'industrie du béton et la construction des préfabriqués en particulier.

Dans la littérature, l'application du concept de produits intelligents pour les bâtiments, nommé par les termes Smart Home (Harper, 2003) ou Smart Building (Morvaj et al., 2011),

met en œuvre une multitude de capteurs, contrôleurs et actionneurs destinés à améliorer le confort des usagers (luminosité, chauffage, climatisation, humidité, etc.) ou encore à assurer leur sécurité. Les travaux de cette thèse visent à réutiliser ces capteurs sans fil, enfouis dans l'infrastructure et les murs du bâtiment, pour mieux contrôler le cycle de vie des éléments composant l'habitation (les préfabriqués). La figure 1 montre ce principe qui est en phase avec les nouvelles idées défendues pendant la réunion à Davos en 2016 par la fondation Ellen Mac Arthur et le Forum économique mondial. Les discussions ont porté sur l'intérêt d'utiliser l'Internet des Objets dans le contexte d'une économie circulaire pour atteindre l'objectif ambitieux du recyclage infini (Ellen MacArthur, 2016). Rendre la structure d'un bâtiment totalement communicant a été imaginé par (Bohn et al., 2005) qui énonce que « *the walls could have ears and these walls could talk* ». Cette idée ouvre un champ de recherche considérable dans l'industrie du béton et la construction d'édifices, tant d'un point de vue technologique que scientifique. C'est dans ce contexte que cette thèse a été initiée.

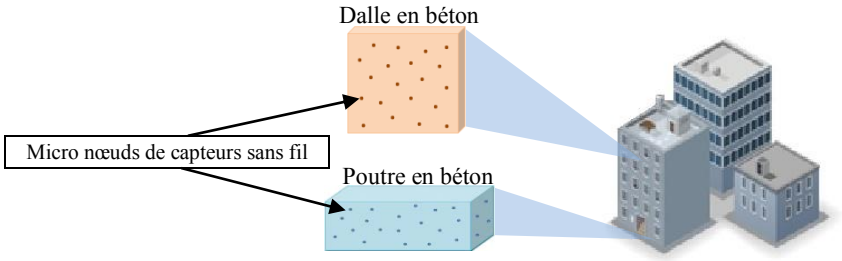


Figure 1. Transformation du bâtiment en objet communicant par application du concept de matière communicante

Notre travail s'intéresse donc au stockage et à la lecture des informations dans des réseaux enfouis dans la matière (en particulier dans un bâtiment) tout au long de son cycle de vie. La contribution scientifique de cette thèse consiste à étudier et à développer des solutions de stockage et de lecture de l'information dans un réseau de micro nœuds de capteurs.

Pour mieux comprendre les contraintes s'appliquant sur les protocoles à développer, il faut analyser les différentes étapes du cycle de vie d'un bâtiment, à savoir : matière première, matière transformée, matière utilisée, matière en maintenance et matière recyclée. Dans les deux premières étapes, la matière subit des transformations sur sa forme (par exemple, la coupe, le séchage, le sciage ou le forage) pour construire le bâtiment. Des opérations de stockage lors de la production sont réalisées afin d'intégrer dans la matière communicante des informations techniques du produit, des informations de caractéristiques matérielles et des informations de traçabilité. Au cours du reste du cycle de vie (troisième, quatrième et cinquième étapes), d'autres opérations de stockage et de lecture sont susceptibles d'être exécutées, en particulier pour optimiser et gérer efficacement les phases de maintenance, les réutilisations possibles des matériaux, ou encore le recyclage. La matière doit donc être capable de stocker/émettre tout au long de sa vie des informations qui lui sont propres.

Parallèlement, le béton peut subir des transformations physiques (par exemple une cloison endommagée, une pièce ajoutée/éliminée, etc.), ce qui peut entraîner des pertes de données si on ne prévoit pas de protocoles de réplication de données dans la matière. Par conséquent, les informations enregistrées dans un préfabriqué doivent être stockées dans la matière de manière uniforme (l'information doit être présente dans chaque partie de la matière). Grâce à cette dissémination uniforme, les informations peuvent être recueillies plus tard par l'utilisateur sans visiter de nombreux nœuds dans la matière, ce qui présente aussi l'avantage d'économiser l'énergie des nœuds sans fil, ressource critique du fait que les capteurs sont enfouis dans la matière.

Cette thèse a ainsi mené aux développements de nouveaux protocoles de communication, adaptés à ce contexte particulier, pour lire et écrire des informations dans l'infrastructure du bâtiment composée de matière communicante. Le protocole de stockage développé a été nommé USEE : *Uniform Storage and Energy Efficient data dissemination protocol* (publié dans (Mekki et al., 2016b)) et le protocole de lecture adapté à USEE s'intitule RaWPG : *Random Walk and Pull Gossip data retrieval protocol*.

Ce rapport de thèse est organisé en quatre chapitres suivis d'une conclusion générale :

Le premier chapitre détaille le concept de matière communicante, qui est une évolution récente du concept de produit intelligent, alliance d'un produit classique (matière inerte) et d'une partie électronique (par exemple, puce RFID), ayant des capacités de communication, de stockage d'informations, et de traitement. Ainsi, après un état de l'art des travaux menés sur les produits intelligents, nous montrons les limites de ce premier concept pour introduire la notion de matière communicante, ainsi que les premiers travaux menés dans ce contexte, en soulignant tout particulièrement ceux ayant été réalisés dans le cadre de la gestion des données de la matière communicante tout au long de son cycle de vie. Pour finir, nous présentons les positionnements technique et scientifique de la thèse.

Le deuxième chapitre commence par un état de l'art sur les réseaux de capteurs sans fil afin de détailler les caractéristiques et contraintes de conception d'une solution dans ce type de réseaux. Nous continuons par la présentation des protocoles de stockage et de lecture dans les réseaux de capteurs sans fil proposés dans la littérature. Chaque présentation est suivie d'une synthèse afin de discuter la pertinence de mécanismes utilisés par rapport à notre contexte applicatif. Cet état de l'art nous permet de sélectionner les travaux les plus intéressants aussi bien pour le stockage que pour la lecture d'informations et servent dans les chapitres suivants soit de bases de travail, soit de protocoles de comparaison.

Le troisième chapitre introduit nos deux protocoles solutions, USEE pour la phase de stockage et RaWPG pour la phase de lecture. La stratégie de chaque protocole est décrite et

justifiée. Ainsi, les modèles de communication, les entêtes protocolaires, les algorithmes et les échanges de messages sont détaillés.

Le quatrième chapitre s'intéresse à l'évaluation des protocoles USEE et RaWPG. Il est divisé en deux parties : la première traite de l'instrumentation du préfabriqué béton pour en faire une matière communicante, la deuxième concerne l'implémentation des protocoles dans l'outil de simulation Castalia/OMNeT++. En effet, pour pouvoir simuler ces protocoles, il faut d'abord définir les paramètres de simulation répondant à notre cadre applicatif en termes d'informations, de nombre de nœuds de capteurs, de densité d'informations, d'autonomie demandée, etc. Ces diverses contraintes sur les simulations sous Castalia/OMNeT++ nous permettent d'évaluer nos propositions en montrant leurs intérêts et leurs limites par rapport à des algorithmes issus de l'état de l'art réalisé au deuxième chapitre.

Le mémoire de thèse se termine par une conclusion générale dans laquelle nous synthétisons les apports de nos travaux et nous en dégageons les perspectives.

CHAPITRE I

Le concept de matière communicante

I. Introduction

Le concept de produit intelligent consiste à associer à un produit physique un composant électronique afin de lui permettre de communiquer, informer, acquérir, décider et réagir aux stimuli de son environnement. Ces dernières années, un nouveau concept nommé « matière communicante » a complètement changé la perception initiale du produit intelligent puisque l'intelligence n'est plus seulement associée ou juxtaposée au produit mais complètement intégrée dans la matière le composant. Ce concept a été le sujet de différent études et applications s'intéressant en particulier à la traçabilité des produits tout au long de leur cycle de vie.

Ce chapitre présente une description de ce qu'est la matière communicante ainsi que les travaux menés autour de ce concept. Ensuite, l'architecture et l'intérêt d'un système permettant de gérer le cycle de vie d'un produit à travers la matière communicante sont présentés. Ce chapitre se termine en montrant les bénéfices apportés en utilisant le concept de matière communicante dans la gestion des bâtiments, en rendant intelligent ses structures en béton. La description de cette application permettra de mieux faire comprendre l'intérêt de développer de nouveaux protocoles de communication adaptés à ce contexte. La définition de ces nouveaux protocoles correspond à la contribution scientifique de cette thèse.

II. Du produit intelligent à la matière communicante

Nous allons présenter l'évolution des technologies sur les produits intelligents ainsi que leurs limites de façon à mieux comprendre les nouvelles approches s'appuyant sur le concept de la « matière communicante ».

II.1. Produits intelligents : état de l'art

Le terme « produit intelligent » a été évoqué pour la première fois en 1988 par (Ives and Vitale, 1988) dans un contexte de service après-vente. Par définition, un produit intelligent est un objet physique (par exemple, au sein d'une chaîne de production ou logistique) doté d'une capacité de perception (capteur), d'une capacité décisionnelle (unité de traitement) et d'une capacité de communication avec les autres produits (unité de transmission). Un produit peut se définir comme un noyau qui présente un certain type d'intelligence générée par l'interaction Produit-Milieu et Produit-Produit. Cette interaction a été définie dans la littérature par le terme « intelligence ambiante ».

Dans (Norman, 1998), l'intégration de ces produits n'est pas vue seulement pour collecter les informations et pour communiquer mais aussi comme un élément naturel de notre quotidien. L'auteur (Gershenfeld et al., 2004) affirme que la barrière entre le monde numérique et notre monde physique doit être éliminée par l'établissement d'un système multipliant les objets interagissant entre eux. Cette vision a été nommée l'Internet des Objets (en anglais : Internet of Things).

Selon (McFarlane et al., 2003), un produit intelligent est défini comme une représentation physique et informationnelle d'un objet avec les caractéristiques suivantes :

- il possède une identification unique,

- il est capable de communiquer efficacement avec l'utilisateur, l'environnement ou avec d'autres produits,
- il peut mémoriser ou stocker des données,
- il dispose d'un langage de communication pour transmettre ses caractéristiques et ses besoins au cours de son cycle de vie,
- il est capable de participer, de prendre des décisions et d'agir sur son environnement (concept d'actionneur intelligent (Zheng, 2014)).

L'identification automatique implique d'obtenir automatiquement l'identité d'un objet au moyen d'une technologie donnée. Par exemple, l'accès en temps réel à l'identité d'un produit grâce à l'identification automatique par radiofréquence RFID permet de déterminer la présence d'un produit et sa localisation (Finkenzeller, 2003). Dans la définition de produit intelligent il est possible de distinguer deux niveaux de complexité : un produit apportant de l'information dans son environnement et un produit supportant des mécanismes de décision (Wong et al., 2002). Ce dernier implique que le produit doit avoir une capacité d'analyse intégrée afin d'évaluer et de prendre la meilleure décision en fonction de son état et de son contexte.

Comme exemple concret de l'application du concept de produit intelligent, nous pouvons citer la fonction de traçabilité (Jansen-Vullers et al., 2003) des produits durant son cycle de vie. Le but dans ce cas est d'enregistrer et d'actualiser toute information dynamique associée à un produit (tels que ses états, les opérations qu'il a suivi, etc.) directement sur un composant électronique (par exemple une puce RFID ou un nœud de capteurs sans fil). Grâce à cela, il est possible qu'un acteur de la chaîne logistique puisse connaître à tout moment l'histoire détaillée d'un produit. Eventuellement, l'information enregistrée sur le composant électronique peut être employée comme une entrée d'un processus afin d'optimiser une opération donnée.

Comme autre exemple, nous citons l'apport de l'information stockée sur un produit intelligent dans un processus de contrôle d'un système de production (McFarlane et al., 2003). L'introduction d'un système d'identification automatique permet au produit physique d'être reconnu et peut ainsi apporter ses informations afin d'influer sur le processus décisionnel.

L'introduction de l'identification automatique fournit la possibilité d'employer l'information concernant chaque produit de façon individuelle (son identité, ses états, son histoire, etc.) en même temps que les données opérationnelles (telles que la température, position, vitesse, statuts des équipements, etc.) afin de contrôler le système. La combinaison de ces deux mécanismes peut améliorer les processus de décision et de contrôle (McFarlane et al., 2003).

La partie décisionnelle associée au produit intelligent permet de prendre des décisions autonomes et ainsi devenir une entité active influençant son cycle de vie (Zouinkhi et al., 2011). Comme avantages de cette approche nous pouvons citer la potentialité d'un produit de s'adapter face aux opérations aléatoires (commandes urgentes, pannes des machines, ruptures de stock, reconfiguration des lignes,

etc.) ou aux réorganisations de la production et du stockage, et aux besoins dynamiques et potentiellement évolutifs des clients.

II.2. Bénéfices générés par les produits intelligents

La mise en place de systèmes basés sur le concept de produit intelligent a plusieurs bénéfices, à savoir :

- gérer les informations produites durant l'ensemble du cycle de vie,
- participer à la synchronisation des flux physique et informationnel,
- fournir en temps réel une visibilité de la chaîne d'approvisionnement,
- améliorer les processus logistiques (fabrication, routage des produits, maintenance, etc.),
- assurer la pérennité des informations (p.ex. lorsque celles-ci sont adjointes au produit via un agent embarqué),
- assurer un niveau de sécurité (p.ex. éviter des erreurs liées à l'intégrité physique ou encore gérer des droits d'accès aux informations).

Les bénéfices économiques proviennent d'autre part, d'une prise en main des informations produit plus rapides et plus sûres. En effet, l'automatisation des données d'entrée réduit le taux d'erreurs (Sahin, 2004). Dans un environnement difficile comme celui d'une chaîne de livraison, la lecture peut être bien plus sûre et rapide en utilisant une technologie du type RFID (comparativement à la solution des codes à barres). Les étiquettes électroniques de type RFID permettent une lecture rapide, une lecture de plusieurs étiquettes de manière quasi simultanée, et des lectures distantes (Thomas, 2009).

En effet, le système de codage manuel de l'information utilisé sur les différentes lignes de production, était sujet à de nombreuses erreurs et par conséquent entrainait de fortes pertes financières. De nombreux exemples de ce genre peuvent être cités. (Chao et al., 2007) énumèrent une multitude de cas industriels, hospitaliers, pharmaceutiques, militaires, environnementaux et mettent en avant les bénéfices atteints pour chacun d'eux. (Tajima, 2007) explique que les gains possibles liés au déploiement de produits intelligents se situent à plusieurs niveaux sur la chaîne d'approvisionnement et se répercutent sur chacun des acteurs (bénéfices du point de vue des acteurs et de la chaîne logistique).

II.3. Limites de concept de produit intelligent

La section précédente a mis l'accent sur les bénéfices pouvant être engendrés par la mise en œuvre de produits intelligents. Néanmoins, un certain nombre de problématiques concernant les différentes solutions et technologies des produits intelligents subsistent encore, comme le soulignent (McFarlane et al., 2012) (Thomas, 2009). Ces derniers déclarent qu'il y a encore un grand nombre de questions ouvertes comme la collecte des informations liées au produit, leur stockage à travers la chaîne logistique, ou encore la dissémination et la gestion de ces informations tout au long de leur cycle de vie. Dans la littérature, les produits dits « intelligents » sont des produits « discrets ». Par exemple, un produit embarque le plus souvent une puce RFID unique afin d'accéder à l'information déportée sur le

réseau. Il peut aussi embarquer un nœud de capteurs afin de réaliser des mesures sur son environnement. Par conséquent, cette notion d'intelligence discrète est limitée, à savoir :

- Lecture discrète : il est parfois nécessaire de « lire le produit » suivant un positionnement particulier, bien que la technologie RFID ait résolu en grande partie ce problème principalement lié à la technologie des codes à barres. Ce qui ne diffère cependant pas entre ces deux technologies, c'est le fait que l'étiquette RFID ou papier est localisée à un endroit particulier du produit, rendant de fait l'information discrète (l'information ne peut pas être attachée à des morceaux/zones différent(e)s de matière durant le cycle de vie de produit).
- Risque d'endommagement du composant électronique attaché : si les composants (puces RFID ou nœuds de capteurs) sont endommagés, le flux d'informations attaché au produit sera dès lors rompu.
- Problème de transfert de l'information : lorsqu'un produit subit une transformation (p.ex. le produit est coupé en deux sous-ensembles ou pièces), une des parties résultantes de l'opération est alors vierge d'informations (excepté si l'on ajoute, après la découpe, un composant électronique à cette nouvelle pièce),
- Faible capacité mémoire : la capacité mémoire actuelle des tags RFID et des nœuds de capteurs est limitée et souvent corrélée à la dimension du composant (plus le tag ou le nœud est petit, plus la capacité mémoire décroît).

II.4. Le nouveau paradigme de matière communicante

Vu les limites du concept de produit intelligent précédemment présentées, un nouveau paradigme a été proposé par (Thomas, 2009) qui change radicalement la manière de voir le produit et la matière. Ce nouveau concept consiste à donner la faculté au produit d'être intrinsèquement et intégralement communicant. « Intrinsèquement » signifie que le produit est par essence communicant, d'un point de vue purement conceptuel, toute la matière est communicante (c'est-à-dire capable d'envoyer/recevoir des informations depuis tout point du produit). Ce concept conduit à un changement important dans le domaine de l'Internet des Objets car on ne se soucie pas de la localisation des composants de communication (p.ex. les tags RFID, les nœuds de capteurs) dans le produit. Ceci peut être illustré par l'image suivante :

« Si on prend un morceau d'acier, lequel est composé de fer et de carbone dans un type de structure donné et qu'on le divise en morceaux de plus en plus petits, nul besoin de se soucier de savoir où sont les atomes de carbone, il y en a partout ! Si ces atomes de carbone étaient les instruments de communication, nous dirions que la matière est intrinsèquement communicante ! ». Ici l'idée du concept de matière communicante est poussée à l'extrême en considérant que l'intelligence puisse être déportée au niveau cellulaire.

Cette vision est loin d'être réaliste à l'heure actuelle à cause des limitations technologiques. Cependant, certaines recherches semblent être prometteuses, tel que le prototype de textile

communicant développé au Centre de Recherche en Automatique de Nancy, dans lequel des µtags RFID (de dimensions 0.15x0.15mm) ont été intégrés en masse dans un textile (Kubler et al., 2011).

Le concept de matière communicante introduit ci-dessus ouvre un champ de recherche considérable, tant d'un point de vue technologique que scientifique. Bien que les technologies actuelles ne permettent pas de rendre l'«atome matière» communicant, la conception de nouvelles solutions, méthodes fournissant de nouveaux services aux utilisateurs ont été proposés pour le futur. Par exemple, un produit fait de matière communicante pourrait servir à mettre en œuvre des solutions de sauvegarde, de copie et de redondance d'informations afin d'améliorer la robustesse de l'accès aux informations embarquées et améliorer la traçabilité de produit.

II.5. Premiers travaux menés dans le contexte de matière communicante

Dans ce qui suit, nous présentons des travaux visant à appliquer le concept de matière communicante, principalement pour des objectifs d'auto-mesure et de traçabilité des produits.

II.5.1. Auto-mesure des produits intelligents

(Krommenacker et al., 2013) et (Peña et al., 2011) s'intéressent à la mesure des dimensions spatiales d'un produit intelligent, dans lequel une multitude de nœuds de capteurs est déployée en surface ou dans le volume du produit. Les auteurs proposent un principe d'auto-mesure, nommé SMS (SMS: Self-Measurement System), où le réseau de capteurs localise dans un premier temps ses nœuds situés en bordure de produit et, dans un deuxième temps, estime les distances entre les nœuds de bordure. Ces nœuds de bordure étant situés sur l'enveloppe du produit, ces différentes mesures fournissent alors des informations sur les dimensions du produit (figure 2).

La détection d'un nœud de bordure est basée sur l'énumération du nombre de nœuds avec lesquels il peut communiquer directement, autrement dit sur la densité de son voisinage. Deux variables QT et QR sont utilisées au cours du processus de détection. QT est la quantité de nœuds théorique dans le voisinage d'un nœud. Elle est calculée en prenant en compte le rayon du champ de transmission radio d'un nœud dans le produit et la densité de déploiement du réseau. QR est la quantité de nœuds réelle dans le voisinage, calculée indépendamment par chaque nœud dans le produit. Chaque nœud doit déterminer et comparer les valeurs QR et QT. Un nœud qui existe au milieu du produit a une quantité théorique égale à la quantité réelle de nœuds ($QR \approx QT$). Par contre, pour un nœud à la bordure du produit, la quantité réelle est largement inférieure à la quantité théorique ($QR < QT$). Une fois les nœuds de bordure détectés, la dernière étape consiste à estimer la distance entre eux afin d'obtenir les dimensions du produit. Cette estimation peut être basée sur les techniques RSS (Received Signal Strength) ou sur le temps d'arrivée ToA (Time of Arrival).

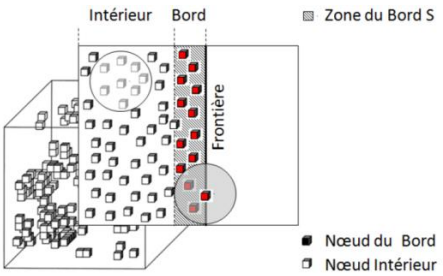


Figure 2. SMS: Self-Measurement System (Krommenacker et al., 2013)

II.5.2. Auto-mesure SMS dans un nanoréseau de capteurs sans fil

Les travaux présentés par (Tran-Dang et al., 2014) visent à mesurer les dimensions d'un produit embarquant dans sa matière des nano-nœuds de capteurs distribués uniformément sous forme d'une grille de dimension 100 cm x 100 cm qui communiquent dans la bande Terahertz (0.1-10 THz). Ces travaux détectent les nœuds de bordure et les nœuds au coin du produit en se basant sur la même technique de densité de voisinage SMS.

Trois types de nœuds *B1*, *B2* et *B3* sont identifiés dans le produit. Ces derniers représentent respectivement les nœuds avec la première, deuxième et troisième densité minimale des nœuds voisins (dans un réseau de capteurs sans fil statique, deux nœuds sont appelés voisin s'ils peuvent communiquer directement l'un avec l'autre) comme l'illustre la figure 3. Les nœuds *B1* sont des coins, les nœuds *B2* sont des nœuds de bordure et les nœuds *B3* sont à l'intérieur du produit.

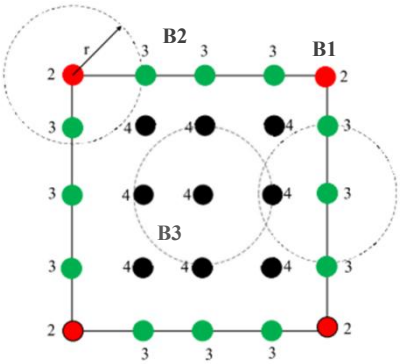


Figure 3. Auto-mesure SMS dans un nanoréseau de capteurs sans fil (Tran-Dang et al., 2014)

Puisque les techniques d'estimation de distance utilisées dans SMS (RSS et ToA) sont difficiles à appliquer dans les nano réseaux de capteurs, les auteurs proposent d'autres algorithmes d'estimation de distance basés sur le comptage du nombre de sauts entre les nano-nœuds *B1* nommés *FBHC* (Flooding-Based Hop-Counting) et *CBHC* (Cluster-Based Hop-Counting) implémentés dans NS-3. Pour *FBHC*, un nœud de coin *B1* génère un paquet et le transmet à d'autres *B1* à travers les nœuds *B2*. Après avoir reçu les paquets, les *B1* récepteurs estiment la distance qui les sépare de nœuds *B1* émetteurs. L'estimation de distance se base sur le rayon *r* du cercle du champ de transmission radio du nano-nœud (les auteurs supposent que tous les nœuds ont le même rayon de champ de transmission radio 1 cm) et le nombre de sauts *HC* réalisé par le paquet au cours de son acheminement à travers les nœuds *B2*. La dimension du produit est égale à *r x HC*.

L'algorithme *CBHC* utilise la technique de « clustering ». Tous les nœuds *B1* et *B2* sont regroupés en clusters. Chaque groupe a des membres du cluster (*CM* : Cluster Members) et un seul nœud chef de cluster (*CH* : Cluster Head). L'algorithme *CBHC* est mis en œuvre en deux phases. La première est la phase de création de clusters et le choix de *CH* pour chacun. La deuxième phase consiste à transmettre un paquet d'un nœud *B1* vers un autre à travers les *CHs*. La distance est estimée en calculant le nombre de sauts réalisés à travers ces *CHs* de la même manière que *FBHC*.

II.5.3. Textile communicant

Les auteurs dans (Kubler et al., 2012a) (Kubler et al., 2012b) (Kubler et al., 2014a) développent de la matière communicante (nommée *e-textile*) obtenue en dispersant régulièrement une grande quantité de µtags RFID (1500 µtags/m²) dans un textile. Les tags utilisés sont des µtags Hitachi ayant une taille de 0.4×0.4×0.1mm. Les tags RFID Omron V720-D52P03 de diamètre 15mm sont aussi utilisés dans ce processus de construction de textile communicant. En outre, le système implique un lecteur RFID connecté à une base de données relationnelle qui contient toutes les informations gérées durant le cycle de vie de produit comme l'illustre la figure 4. À chaque opération d'écriture, la base de données est explorée pour sélectionner les données pertinentes (lignes des tables de base de données) qui doivent être stockées dans la matière. Un niveau d'importance *I* entre 0 et 1 est affecté à chaque élément de donnée sélectionné. Ce niveau est calculé par un algorithme d'évaluation de la pertinence des informations basé sur *Fuzzy-AHP* (Kubler et al., 2014b). Les éléments de données avec les plus hauts niveaux d'importance sont ensuite stockés dans les tags lorsque le textile passe sous un lecteur RFID dont le rôle est d'écrire mais aussi de lire les données. Comme la technologie RFID est limitée en mémoire, l'information est découpée en différents segments et distribuée sur plusieurs tags RFID (chaque tag stocke un segment) à l'aide d'un entête protocolaire spécifique qui permet également de reconstruire l'information initiale dans une phase de lecture. Cette application est détaillée dans la suite de ce rapport puisqu'elle présente une architecture référence d'un « système matière communicante » pour la gestion du cycle de vie d'un produit.

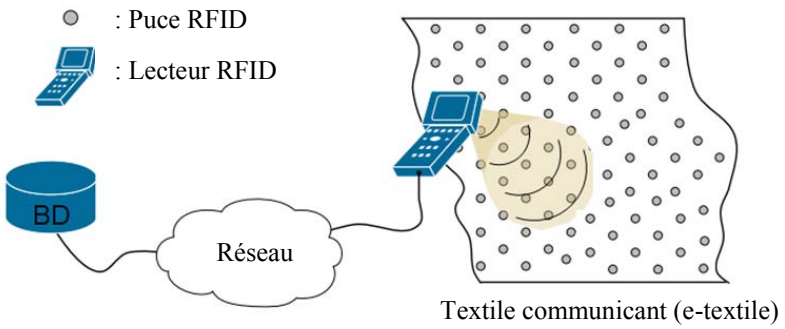


Figure 4. Processus de dissémination de données dans un textile communicant

II.5.4. Traçabilité du bois par marquage chimique

Les travaux présentés par (Jover et al., 2011) (Jover, 2013) (Jover et al., 2013) présentent une solution de traçabilité du bois qui permet de conserver l'information relative à son origine tout au long de son

cycle de vie. La solution proposée repose sur le fait de pouvoir attacher au bois d'un arbre dans son intégralité une marque qui lui soit propre. Ce marquage dans la masse pourrait permettre de connaître, tout au long de son cycle de vie, les informations le concernant. Les auteurs ont proposé une approche par marquage chimique du bois en noyant le billon dans un assemblage de plusieurs produits chimiques identificateurs du producteur, et une méthode de spectroscopie nommée RQN (RQN : Résonance Quadripolaire Nucléaire) pour identifier ce marquage. Ces études montrent la faisabilité de l'identification d'un produit chimique imprégné et recristallisé dans un échantillon de bois et prouvent qu'il est possible de marquer l'origine du bois dans la masse avec un assemblage de plusieurs produits chimiques. Cependant, dans cette application, la seule information embarquée dans la matière est la référence du produit réalisée par le marquage chimique. Le produit ne porte donc pas toute son information.

III. Gestion de cycle de vie d'un « système matière communicante »

La nécessité de gérer les informations du produit durant l'ensemble de son cycle de vie est devenue incontournable, d'une part, en raison des techniques de sophistication des produits et, d'autre part, en raison des régulations gouvernementales strictes concernant la gestion du cycle de vie des produits. Aujourd'hui, étudier les cycles de vie des produits fait partie intégrante des stratégies des entreprises pour concevoir, gérer et planifier au mieux leur vie. Des systèmes de gestion ont depuis vu le jour, principalement du fait de l'expansion sans précédent des systèmes et outils informatique/d'information. Ces systèmes de gestion se regroupent sous l'acronyme PLM (Product Lifecycle Management).

III.1. Le concept PLM

L'acronyme PLM a émergé dans les années 1990 et s'est formé autour de deux courants (Lee et al., 2008) (Rachuri et al., 2008), à savoir :

- La gestion des entreprises : elle se compose de systèmes de planification des ressources matérielles et de l'entreprise (ERP - Entreprise Ressource Planning), de systèmes de gestion des relations clients (CRM - Customer Relationship Management) et de gestion de la chaîne logistique (SCM- Supply Chain Management).
- La gestion des « informations produit » : de nombreux aspects/secteurs de l'entreprise sont connectés à la vie du produit, ce qui génère de grandes quantités d'informations à stocker, à échanger et à gérer.

Le cycle de vie d'un produit que (Kubler, 2012) décrit, est découpé suivant les processus dans lesquels le produit transite durant sa vie. La figure 5 présente un découpage du cycle de vie du produit suivant six processus : conception, fabrication, entreposage, commercialisation, utilisation et recyclage. Chaque processus est constitué, à son tour, de sous-processus ou d'activités qui sont nécessaires, par exemple, durant la transformation et/ou l'utilisation du produit. Un produit dans un système PLM n'est

pas nécessairement un produit physique, cela peut être un service ou tout autre produit incorporel (p.ex. un logiciel).

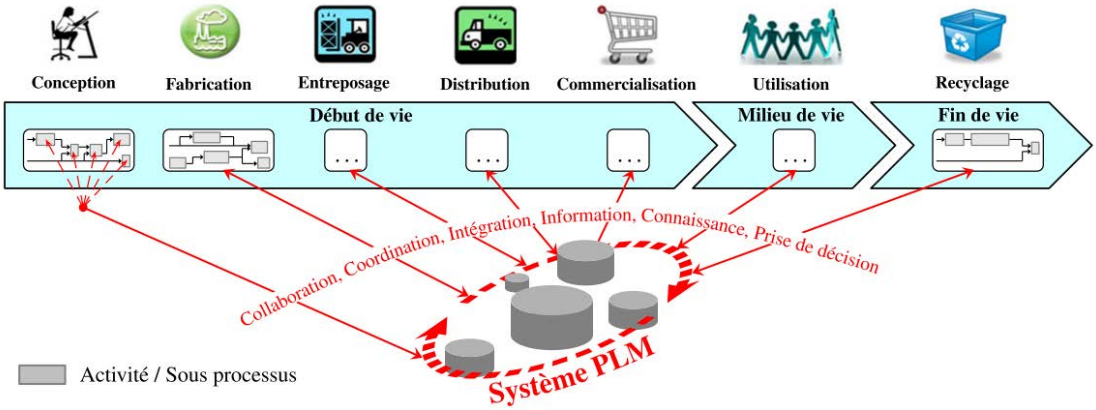


Figure 5. Vue générale du système PLM et ses objectifs (Kubler, 2012)

Le système PLM gère l'intégration d'une multitude de systèmes, d'applications, d'outils et de bases de données pour gérer les informations :

- Des applications utilisées pour gérer les informations techniques comme les systèmes TDM (Technical Data Management),
- Des applications gérant les informations d'exécution de fabrication comme les systèmes MES (Manufacturing Execution System),
- Des applications gérant les informations relatives aux ordres de maintenance comme les systèmes CMMS (Computerized Maintenance Management System),
- Des outils permettant d'optimiser les activités de la chaîne logistique et des entreprises à un instant donné, ou à travers le temps comme les systèmes APS (Advanced Planning Systems),
- Des outils supportant les opérations de fabrication, de planification des processus et des méthodes de conception, tels que les systèmes CAPP (Computer Aided Process Planning).
- Des outils permettant d'évaluer l'impact d'un produit sur l'environnement depuis sa conception jusqu'à son démantèlement, comme les outils LCA (Life Cycle Assessment).

Tous ces systèmes fournissent des fonctions dédiées au domaine d'application, ainsi que les bases de données qui leur sont associées (contenant les informations propres à l'entreprise) (Jaber, 2009).

III.2. Architecture PLM de « système matière communicante »

La première architecture définissant un système PLM relatif à la matière communicante a été présentée par (Kubler, 2012). Les informations générées par les applications relatives au système PLM sont principalement stockées dans des bases de données. Ces bases de données sont formalisées par des modèles de type entités-relations et leur mise en place nécessite des techniques particulières, permettant d'optimiser les performances du système (gain de temps pour accéder aux informations, respect des exigences relatives à la cohérence et à la disponibilité des informations, etc.). Cette étape correspond au *niveau 1* de l'architecture du système d'information relatif aux « système matière communicante », comme illustré sur la figure 6.

Le *niveau 2* de l'architecture concerne la dissémination des informations dans la matière des produits communicants. Les informations pertinentes pour les acteurs du cycle de vie sont disséminées et répliquées (si souhaité/nécessaire) directement sur les produits communicants, comme illustré sur la figure 6 (voir flèches vertes en pointillés).

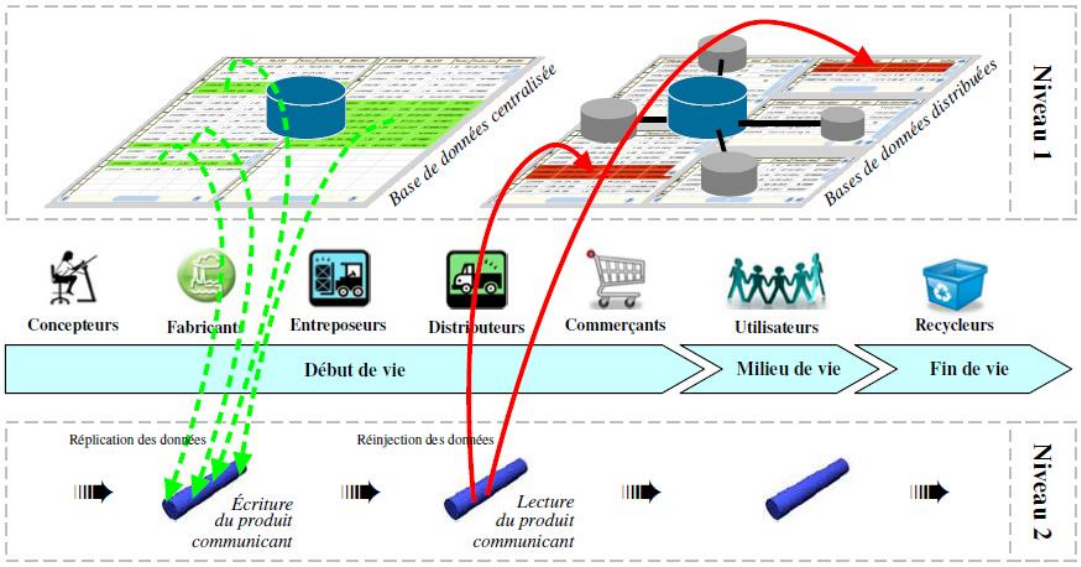


Figure 6. Système d'information à deux niveaux lié au « système matière communicante » (Kubler, 2012)

Dans cette architecture, la matière joue uniquement le rôle de vecteur d'informations, capable de véhiculer durant son cycle de vie les informations entre acteurs et systèmes d'information. La matière communicante véhicule exclusivement des données qui lui sont propres, qui auraient été, par exemple, trop coûteuses et/ou impossibles de véhiculer d'une autre manière (p.ex. mise en place des accès entre fournisseurs trop coûteux ou impossibles), des informations de traçabilité, ou encore des informations destinées à offrir de nouveaux services aux utilisateurs (p.ex. des professionnels capables de communiquer avec les matériaux afin de surveiller leur état interne). Ces données sont par la suite lues et extraites de la matière communicante (à un point de lecture donné) et peuvent être directement exploitées/réinjectées dans le système de bases de données de l'utilisateur. Cette étape de « réinjection » ramène les données récupérées au *niveau 1* de l'architecture comme le montrent les flèches rouges en trait plein dans la figure 6, où des mécanismes de mises à jour des informations peuvent être employés.

Dans ce système d'information, (Kubler, 2012) a plus particulièrement étudié le transfert des informations depuis le *niveau 1* vers le *niveau 2*, appelé *processus de dissémination*. Les travaux présentés dans cette thèse étant fortement couplés avec le processus de dissémination, nous allons le détailler dans la prochaine section.

III.3. Processus de dissémination d'informations dans un « système matière communicante »

Le processus de dissémination d'informations dans la matière communicante consiste à sélectionner des informations pertinentes dans une base de données (sélectionner des informations à stocker dans le

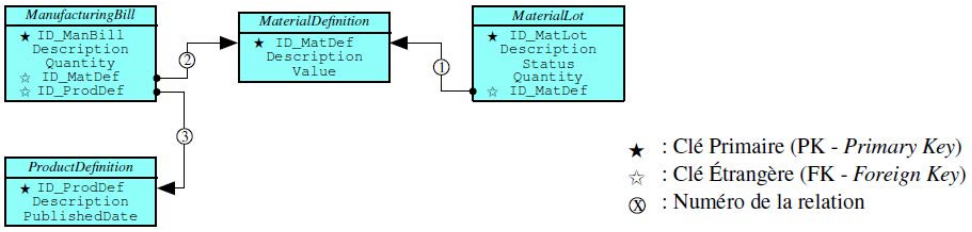
produit) située dans le *niveau 1* pour les stocker dans un produit composé de matière communicante, situé au *niveau 2* du processus du système d'information. Ce processus est une séquence de plusieurs étapes détaillées ci-après.

III.3.1. Identification et évaluation de la pertinence des « informations produit »

Lorsque le produit se présente à un point d'écriture durant son cycle de vie, cette étape vise à identifier au sein des bases de données toutes les informations qui sont liées au produit, puis à évaluer leur pertinence à être (ou ne pas être) répliquées/embarquées sur le produit communicant.

a) Identification des « informations produit » au sein de la base de données

Il s'agit dans un premier temps de réaliser un filtrage des informations afin d'éliminer celles n'ayant pas d'intérêt à être stockées/répliquées sur le produit et par conséquent, pas d'intérêt à être évaluées. L'identification des « informations produit » est effectuée à l'aide d'un algorithme qui prend en entrée la structure du Modèle Logique de Données (MLD) ainsi que les données des tables. La figure 7(a) présente un exemple d'une partie d'un MLD et des relations clés primaires (PK - Primary Key) - clés étrangères (FK - Foreign Key) étudié par (Kubler, 2012). Une fois implémenté, ce MLD conduit aux tables de la base de données comme illustré dans la figure 7(b) avec comme exemple une table nommée *MaterialDefinition*. Sur l'exemple de la figure 7(b), seul le tuple 3 est identifié via l'algorithme d'identification comme étant un « tuple produit » (les tuples 1, 2 et 4 ne sont aucunement liés au produit).



(a) Exemple d'une partie d'un Modèle Logique de Données (MLD)

Indique un « tuple produit » (lié à d_p)

	1 (★) ID_Material	2 Description	3 Value
1	MD041...	Wooden plank with a nominal...	$4m$ of...
2	MD993...	Textile with a high developed pol...	$3mm$...
3	MD060...	Textile which is provided with...	15°C
4	MD770...	Vehicle headrests that conform...	$2 \times$ ...

(b) Table *MaterialDefinition* et les « tuples produit » identifiés

Figure 7. Vue d'une partie du MLD de la table *MaterialDefinition* et des « tuples produit » (Kubler, 2012)

L'approche développée par (Kubler et al., 2014a) propose, lorsque le produit se présente à un point d'écriture, d'extraire l'ensemble des tables T et de les parcourir à l'aide d'un algorithme particulier, nommé *RetrievalData*. Cette approche prend en entrée la référence portée par le produit qui est utilisée comme pointeur vers une (ou plusieurs) information(s) au sein de la base de données. Cette référence produit est notée *dp* (abréviation pour Data item Product). L'algorithme *RetrievalData* identifie tous les tuples qui sont liés d'une manière ou d'une autre à la référence *dp*, en parcourant une à une

l'ensemble des tables *T*. Le principe de l'algorithme est de parcourir les tables en s'appuyant sur la connaissance du modèle, c'est-à-dire sur la connaissance des clés primaires (PK) et clés étrangères (FK). *RetrievalData* identifie les clés étrangères d'une table donnée (celle en cours d'exploration), puis poursuit son exploration dans les tables correspondant à ces clés étrangères (où FK est PK). Ainsi, au cours de cette exploration, de nouveaux « tuples produit » sont identifiés.

b) Evaluation de la pertinence des « informations produit »

(Kubler et al., 2014b) propose de calculer, pour chaque information identifiée précédemment, un indice représentant l'intérêt de l'embarquer sur la matière (appelée *pertinence* de l'information). La figure 8 reprend à l'identique la table *MaterialDefinition* décrite précédemment et affiche la pertinence des trois data items issus du tuple 3. Le data item localisé en colonne 1 et ligne 3 de *MaterialDefinition*, noté $T_{MatDef}\{3, 1\}$, possède à un instant de cycle de vie de produit une pertinence de 0.33. Plus cette pertinence est grande, plus le data item est prioritaire pour être embarqué sur le produit. Elle est comprise entre 0 (information à faible valeur ajoutée, peu importante) et 1 (information à forte valeur ajoutée, très importante).

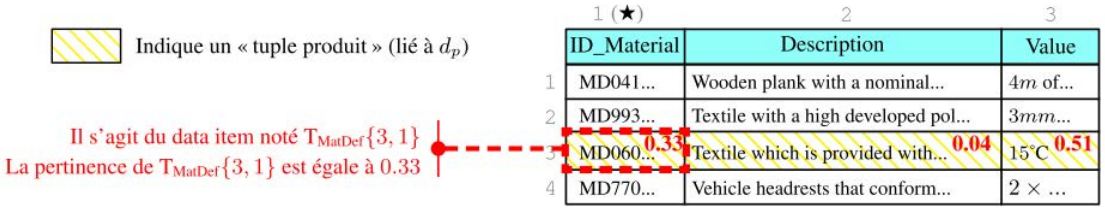


Figure 8. Table *MaterialDefinition* et pertinence des data items issus des « tuples produit » (Kubler, 2012)

Le mécanisme de calcul de la pertinence, qui est le cœur des travaux de (Kubler, 2012), est basé sur une approche *AHP-Floue* et sur l'utilisation conjointe de différents critères. L'approche AHP (Analytic Hierarchy Process) permet de prendre en compte des critères aussi bien qualitatif. Couplée à la théorie des sous-ensembles flous, elle permet en outre de représenter l'information en tenant compte de sa complexité et des divers points de vue des experts.

III.3.2. Stockage des données pertinentes dans la matière communicante

L'idée développée dans (Kubler et al., 2012b) consiste à concrétiser le concept de matière intrinsèquement et intégralement communicante par la dissémination en masse des µtags RFID au sein d'un textile. Un premier prototype de textile communicant a été conçu (voir image sur la figure 9), qui a été répertorié dans la catégorie des e-textiles.

L'information devant être stockée sur le textile communicant correspond à l'ensemble des data items qui ont été jugés pertinents à un instant précis du cycle de vie de la matière. Puisque les technologies µtags RFID sont contraintes en mémoire, l'information est disséminée/éclatée sur plusieurs tags. La dissémination repose sur un en-tête protocolaire particulier qui permet d'éclater les data items sur l'ensemble des tags présents dans le textile durant une opération d'écriture et par la suite, de les reconstruire durant une opération de lecture. Elle nécessite :

- Une matière communicante capable de stocker des informations (une technologie RFID de type R/W - Read/Write),
- Un lecteur RFID permettant de dialoguer avec les tags présents dans la matière,
- Un accès à la base de données :
  - Durant une opération d'écriture du produit : accès à la liste des data items devant être évalués et stockés sur le textile,
  - Durant une opération de lecture du produit : il n'est pas forcément nécessaire d'accéder à la base de données, excepté si l'utilisateur désire mettre à jour les informations ou compléter les informations portées par le textile.

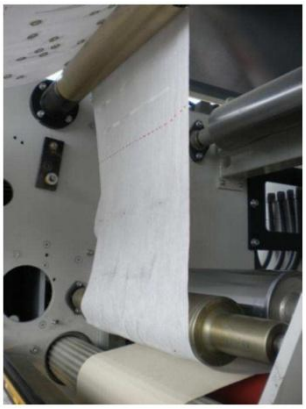


Figure 9. Production d'un textile communicant

Après avoir présenté le concept et les travaux de recherche sur la matière communicante, nous allons maintenant décrire la problématique de cette thèse qui consiste à adapter l'idée de la matière communicante pour améliorer la gestion du cycle de vie d'infrastructures en béton.

IV. Problématique de thèse

Dans cette section, nous allons tout d'abord expliquer notre contexte industriel pour délimiter le périmètre de cette thèse, et ensuite développer notre position à la fois scientifique et technique en regard des travaux présentés dans la section précédente.

IV.1. Contexte

Actuellement, le CRAN mène des études et recherches en collaboration avec le Centre de Recherche pour l'Industrie de Béton (CERIB : www.cerib.com) qui représente 650 entreprises de préfabrication béton en France. Cette thèse vise ainsi à présenter de nouvelles solutions de gestion d'informations dans le domaine de la construction béton via le concept de matière communicante.

Lors de la construction d'un édifice, les entreprises de la construction béton suivent majoritairement deux stratégies :

- *La fabrication sur place* : tous les éléments de la construction sont fabriqués sur place via des coffrages. Une centrale à béton est installée sur le site du chantier. Le site est livré seulement en matière première (granula, ciment, adjuvant, etc.) et en ferrailles.

- *La préfabrication* : tous les éléments de la construction (poutres, planchers, murs porteur, piliers, etc.) sont fabriqués en usine loin du chantier et apportés sur le site pour être mis en œuvre rapidement.

Pour des raisons d'industrialisation et de coût, le choix s'oriente de plus en plus vers la préfabrication. En effet comme il a été décrit dans l'un des quatre rôles du management de la construction (Vrijhoef, 2000) : il faut transférer les activités du site de construction plus tôt dans la chaîne logistique. Cela permet d'éviter les aléas climatiques du site et de mener plusieurs opérations en parallèles qui n'auraient pas pu être faites simultanément sur le site. Il est possible avec la préfabrication de gagner en place, en temps et en qualité.

Différents travaux ont été menés sur l'utilisation du concept de produits intelligents pour améliorer la préfabrication de structures en béton. Dans la suite, différentes applications sont présentées.

IV.2. Premières applications de produits intelligents dans la construction béton

Bien que le paradigme « matière communicante » soit récent, on peut trouver dans l'industrie du béton des initiatives qui s'en approchent.

IV.2.1. La maturométrie

La place du béton dans l'économie en fait un sujet de recherche de prédilection. Ses propriétés sont bien connues et il a été démontré qu'il était possible de connaître la résistance mécanique d'un élément en suivant sa température après coulage (Gauthier and Regourd, 1982). Bien que connue depuis 35 ans, les applications industrielles de cette connaissance sont arrivées :

- D'une part, avec la nécessité de toujours gagner du temps : en effet, par défaut, il faut au minimum 14 jours pour que le béton d'un élément préfabriqué ait les propriétés adéquates pour être mis en œuvre. Le suivi de l'évolution des propriétés permet de valider au plus juste sa mise en œuvre.
- Et d'autre part, le développement rapide des technologies de capteurs : embarquer des capteurs directement dans le béton, pour des coûts abordables, sans modifier ni les propriétés, ni l'aspect extérieur des produits n'est possible que depuis peu.

Par exemple, la société BizzDev pour le compte de Bouygues Construction (Albin, 2014), a créé un système composé de capteurs de températures directement embarqués dans le préfabriqué. Le béton est ainsi capable de délivrer sa température, de façon continue dans le temps.

IV.2.2. La détection de fissures

Le suivi de l'évolution des structures est une problématique importante de la construction béton. La détection de fissures est aujourd'hui encore très visuelle. Des chercheurs ont essayé via des tags RFID placés régulièrement en surface de faire un premier pas vers une détection automatique de l'apparition de fissures (Kalansuriya et al., 2013). Bien que s'approchant du concept de matière communicante par son aspect continu, cette initiative a ses limites puisque la détection de fissures passe par la non-

détection d'un tag par l'opérateur. Pour aller plus loin, il faudrait que la poutre soit elle-même capable de signaler qu'un de ses tags est défectueux et qu'il y a donc probablement une fissure.

IV.2.3. L'identification du béton

Il arrive souvent sur certains chantiers et notamment ceux qui se situent en ville, que le béton soit livré frais dans des camions toupies. La provenance de ce béton doit être connue et ce, même après le coulage de l'édifice. La société Lafarge pour le chantier de la tour D2 à Paris (LAFARGE, 2013) a choisi d'incorporer directement dans ses bétons, avant même qu'ils ne soient coulés, des tags RFID afin de pouvoir les identifier. Après écoulement du béton, les tags RFID sont présents dans l'édifice au nombre de 4 ou 5 tags tous les 2 m^3 en moyenne. La nature du service rendu est limitée, puisqu'il s'agit d'une simple identification des tags RFID et d'une consultation de la provenance sur une base de données exportée comme l'illustre l'image sur la figure 10.

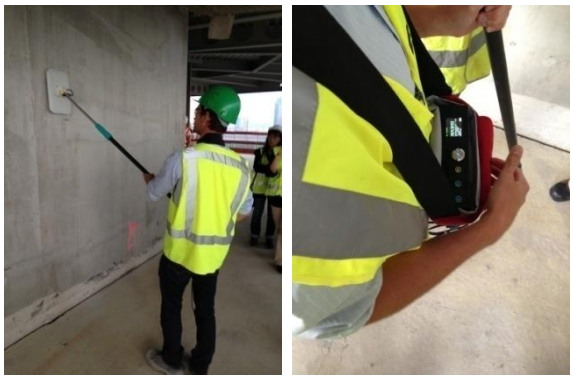


Figure 10. Lecture de l'identifiant des tags RFID par un opérateur

IV.2.4. Marquage CE d'une poutre en béton

À l'image de l'initiative de Lafarge pour le chantier de la tour D2 à Paris, la société CERIB a procédé à l'utilisation de la technologie RFID dans la construction béton pour des problèmes de traçabilité dans la chaine logistique des éléments préfabriqués en béton (Albin, 2014). En effet, le marquage CE impose la transmission d'un certain nombre de documents dont : une étiquette synthétisant les informations concernant le produit, sa fiche de manutention et une déclaration environnementale. Ces documents risquent de se perdre ou d'être endommagés tout au long de chaine logistique. Pour résoudre ce problème de logistique, CERIB a proposé d'utiliser le concept de produit intelligent. Ainsi, des poutres en béton ont été instrumentées en intégrant deux tag RFID à chaque extrémité. Ensuite, les informations CE ont été enregistrées dans la mémoire de ces tags, permettant ainsi d'augmenter les chances que ces informations soient disponibles tout au long de la chaîne logistique. Une première assistance à la gestion des stocks et de leurs emplacements sur le chantier a aussi été mise en place.

IV.3. Intérêts de la matière communicante dans la construction béton

Dans la littérature et comme présenté précédemment, les systèmes préfabriqués sont instrumentés et marqués par l'utilisation de tags RFID et de capteurs. Cela fonctionne bien dans le cas où le produit

fini est de taille réduite (au maximum quelques mètres). Cependant, dans certaines configurations, cela peut poser des problèmes :

- **Si le produit est amené à être divisé.** Cette constatation prend son origine dans le processus de première transformation du bois pour la construction dans lequel un élément initial unique, l'arbre dont la provenance doit être connue, va être divisé en plusieurs produits et à leur tour subdivisés ou transformés. Ainsi un mur, ou un bâtiment, réalisé par assemblage d'éléments préfabriqués pourrait être désassemblé partiellement pour des raisons de maintenance ou de modification voire de démantèlement, par exemple.
- **Si le produit dépasse l'échelle humaine**. Il est fréquent de rencontrer sur certains grands édifices des poutres ou dalles en béton de plusieurs dizaines de mètres, voire d'une centaine de mètres. Il est alors difficile pour l'opérateur de parcourir systématiquement l'ensemble de l'édifice pour identifier ses caractéristiques en recherchant le ou les tags contenant les informations à lire.

La matière communicante est donc une solution possible pour résoudre ces problèmes. Elle ouvre aussi la porte sur la création de nouvelles fonctionnalités, et va donc donner de la valeur ajoutée au préfabriqué. La matière communicante, contrairement au produit intelligent, rend l'information continue dans le préfabriqué car :

- D'une part, l'information est « saupoudrée » dans la matière avec des taux de duplication qui peuvent être fonction de leur importance.
- D'autre part, l'information peut être diffusée (transportée) à travers la matière, ce qui permet :
  - De garantir qu'une information est toujours présente dans le béton (même en cas d'éclatement de celle-ci suite à des opérations de tronçonnage, de perçage, etc.),
  - De permettre à l'opérateur de lire les caractéristiques du béton en tout point de sa structure.

Les informations stockées dans le préfabriqué peuvent être aussi enrichies au cours du cycle de vie de la structure en béton avec des mises à jour faites par l'opérateur ou de façon complètement autonome par des capteurs sans fil intégrés dans la structure qui permettront par exemple d'observer certaines fragilités sur le béton (après un tremblement de terre par exemple) ou de connaitre l'exposition du béton à des polluants/radiations qu'il faudra considérer lors du démantèlement et recyclage de la structure.

La généralisation de la matière communicante non seulement sur le béton mais sur d'autres matériaux du bâtiment (p.ex. le bois) permettra aussi de mettre en coopération tous ces matériaux pour mieux comprendre leurs interactions.

On pourrait même imaginer que le béton communicant permettra aussi aux habitants d'une maison de connaître les propriétés physiques de ses murs pour envisager plus sereinement des travaux de bricolage (bon endroit pour faire un perçage, utilisation d'une colle adaptée au béton, etc.). Le mur en béton pourrait finalement être utilisé comme un disque dur familial enregistrant des photos qui

pourront directement être lues par des écrans collés au mur ou accessible à travers des systèmes LiFi (Light Fidelity) avec des LEDs implantées dans le mur pouvant communiquer avec des téléphones portables ou autres équipements domotiques.

IV.4. Positionnement

Dans ce qui suit, nous présentons le positionnement technique et scientifique de la thèse dans le contexte de développement des préfabriqués en béton par le concept de matière communicante.

IV.4.1. Positionnement technique

L'objectif des travaux de thèse est de transformer les bétons préfabriqués (p.ex. dalles, poutres, planchers, murs, piliers, etc.) en objet communicant en dispersant uniformément des composants électroniques communicants dans le béton.

Deux technologies ont été étudiées : la technologie RFID et la technologie des réseaux de capteurs sans fil. L'intégration d'une multitude de puces RFID dans le béton préfabriqué d'un édifice nécessite une communication directe avec un lecteur RFID pour gérer les informations contenues dans leurs mémoires. En phase d'écriture et lecture des informations, le lecteur RFID doit passer sur toutes les puces de la structure en béton. Cela semble difficile dans une infrastructure de larges dimensions (p.ex. immeuble de grande hauteur). Pour cette raison les réseaux de capteurs sans fil ont été choisis dans nos travaux. L'idée est donc de disperser uniformément des micro-nœuds de capteurs sans fil dans le béton comme cela est montré dans la figure 11. L'intégration de réseaux de capteurs dans les matériaux a été proposée en premier par (Peña et al., 2011), mais seulement pour des objectifs d'auto-mesure de produit intelligent. L'utilisation des réseaux de capteurs permet aussi une capacité de stockage dans la matière plus importante que la technologie RFID. À titre d'exemple, la capacité d'un µtag RFID utilisé dans les travaux d'*e-textile* communicant (Kubler et al., 2011) est de 30 octets, par contre la capacité maximale d'un micro nœud de capteurs peut atteindre 128 Ko (Yamashita et al., 2006). De plus, contrairement à la technologie RFID, une connexion à un seul nœud du réseau de capteurs est suffisante pour atteindre tous les nœuds dans le béton de l'édifice en utilisant des protocoles de communication multi-sauts. Pour terminer, un autre avantage réside dans le fait que les nœuds peuvent embarquer des capteurs, permettant ainsi de rendre la matière communicante active. De nombreux scénarios de maintenance prédictive (p.ex. détection de fissures, instabilité, etc.) pourraient alors être envisagés.

Dans cette thèse, les nœuds de capteurs dans le béton sont donc utilisés comme un vecteur d'informations servant pour :

- Porter les informations telles que celles utiles au marquage CE et éviter les risques de perte d'informations entre les différents acteurs de la chaine logistique (déplacement de produits entre le constructeur de préfabriqués, le transporteur et la maitrise d'œuvre).

- Mettre en œuvre un édifice. Le béton peut en effet contenir des informations sur ses caractéristiques techniques, sur son origine et sa date de fabrication, ou sur ses procédures de maintenance.

Pour illustrer les avantages de notre proposition par rapport aux solutions industrielles basées sur la technologie RFID, citons l'exemple précédemment présenté du chantier de la tour D2 à Paris, réalisé par la société Lafarge. Les tags RFID enfouis dans le béton avec une densité de 4 ou 5 tags tous les 2 m^3 sont seulement des émetteurs. Leur identifiant, qui est unique, permet ensuite de recueillir les informations relatives à celui-ci dans une base de données (p.ex. le numéro du bon de livraison, la centrale d'origine, le nom commercial, la date et l'heure de fabrication du béton, etc.). L'utilisation d'une solution basée sur des réseaux de capteurs permet de stocker toutes les informations relatives au béton dans les mémoires des nœuds, dont les fonctionnalités de mesure peuvent être utilisées pour d'autres services.

Dans le cadre de ce mémoire, la matière communicante considérée est donc constituée d'un matériau dans lequel est enfoui un réseau de micro nœuds de capteur sans fil ayant des capacités de mesure, de communication et de stockage de données.

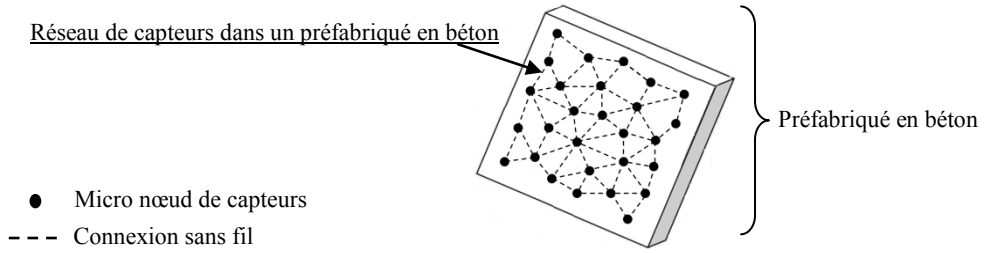


Figure 11. Dispersion régulière et contrôlée des micro-nœuds dans un préfabriqué en béton

IV.4.2. Positionnement scientifique

Une première architecture pour la gestion des informations dans le béton a été proposée par (Albin, 2014). L'architecture comporte trois parties comme le montre la figure 12 :

- Une application web (rectangle en pointillés rouges sur la figure 12) permettant au constructeur du préfabriqué de gérer les informations stockées dans la base de données et aux intervenants de consulter ces informations durant le cycle de vie de la structure en béton,
- Une application intermédiaire (middleware) faisant office d'interface entre le mobile et la base de données (rectangle en pointillés verts sur la figure 12).
- Une application mobile (p.ex. un lecteur RFID) connectée au béton communicant (rectangle en pointillés bleus sur la figure 12). Cette application permet de faire des opérations de consultation et de gestion des informations dans le préfabriqué par le constructeur et tous les intervenants durant son cycle de vie. Ces opérations sont assurées

par des fonctions de stockage et de lecture d'information dans la mémoire des composants électroniques intégrés dans le béton.

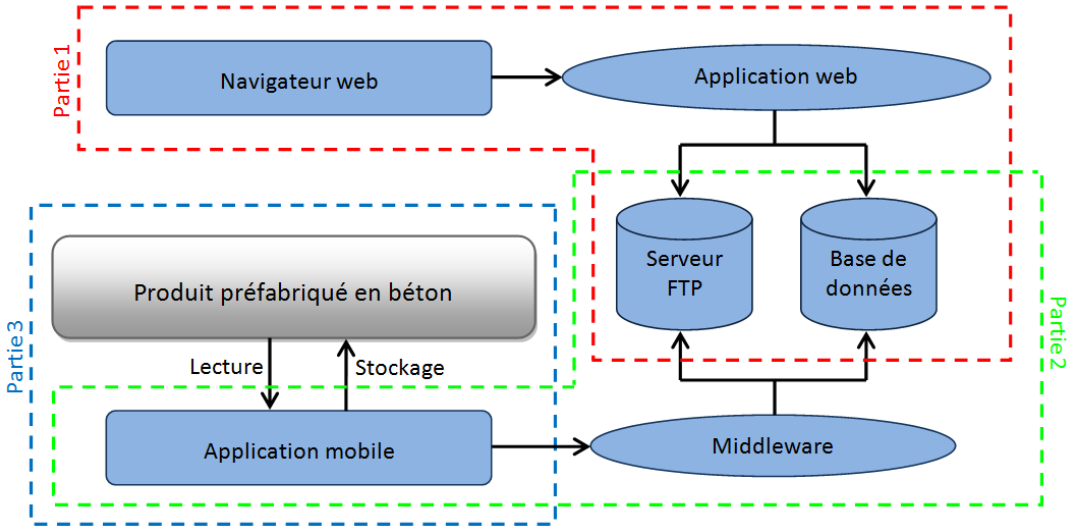


Figure 12. Première architecture pour la gestion des informations dans le béton

Nos travaux se rapportent à la partie 3 de cette architecture. La contribution scientifique de cette thèse consiste à étudier et à proposer des solutions de stockage et lecture d'informations dans le réseau de micro-nœuds de capteurs enfoui dans la matière communicante (dans notre cas d'étude, un préfabriqué béton).

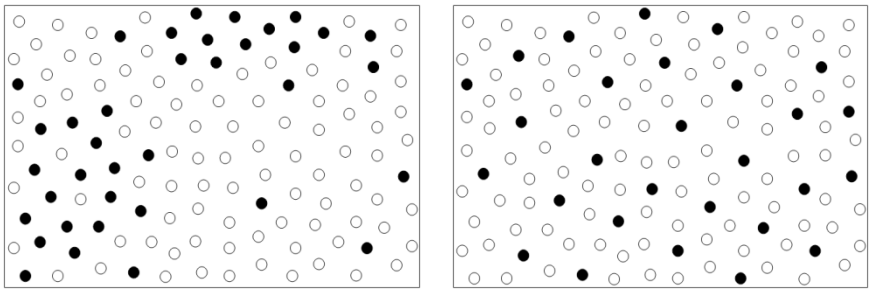
Au début du cycle de vie du préfabriqué, des opérations de stockage lors de la production sont primordiales afin d'intégrer des informations propres au constructeur. Ainsi, les informations des caractéristiques matérielles et de traçabilité sont intégrées dès la phase de production. D'autres opérations de stockage et lecture sont susceptibles d'être exécutées durant tout le cycle de vie du préfabriqué pour optimiser et gérer efficacement les phases de chaine logistique, de maintenance, de réutilisation, ou encore de recyclage au sens de l'économie circulaire (Zaetang et al., 2016) (Ellen MacArthur, 2016).

Ces fonctions de stockage et de lecture dans un réseau de capteurs nécessitent le développement de nouveaux protocoles de communication multi-sauts devant respecter un certain nombre de contraintes énumérées ci-après.

Pour la phase de stockage, le point clé est que l'information soit répliquée de façon uniforme dans le préfabriqué. Comme le montre la figure 13(a), lorsque les informations ne sont pas uniformément répliquées dans le préfabriqué, des endroits sont chargés de données alors que d'autres sont vides (cas à éviter). Par contre sur la figure 13(b), les informations sont uniformément répliquées dans tout le préfabriqué, de sorte que tous les endroits du produit contiennent des données (cas cible). Recourir à une dissémination uniforme présente de nombreux avantages :

- Les informations peuvent être recueillies facilement par un opérateur sans parcourir toute la structure en béton,

- Les informations peuvent être recueillies sans visiter de nombreux nœuds dans le béton, ce qui conserve les ressources énergétiques du reste du réseau de capteurs,
- Les informations sont accessibles même en cas de dégradation partielle de la structure en béton.



(a) (b)

o nœud vide de données ● nœud stockant de données

Figure 13. Uniformité de stockage des informations dans le réseau de capteurs du préfabriqué

Par ailleurs, les informations à stocker peuvent avoir des niveaux d'importance différents. Comme présenté précédemment, l'approche de (Kubler et al., 2014b) cherche à extraire les « informations produit » d'une base de données pour les stocker dans un système matière communicante. Ces « informations produit » sont sélectionnées par niveau d'importance, de sorte que les informations à haute valeur ajoutée pour le cycle de vie seront marquées avec une valeur d'importance plus élevée. Ce niveau d'importance est pris en compte dans nos travaux, et va affecter la densité de réplication de l'information (nombre de nœuds stockant l'information) dans le réseau de capteurs. Par exemple, si l'importance est égale à 1 (information classée très importante), les données doivent être présentes dans tous les endroits du préfabriqué. Les informations moins essentielles vont se retrouver disséminées de façon plus dispersée. Mais, quels que soient le taux de réplication et la densité de stockage de l'information, l'information devra toujours être répliquée uniformément.

En conclusion, le processus de stockage doit répondre à deux contraintes : l'uniformité du stockage et un taux de réplication fonction de l'importance de l'information.

Lors de la phase de lecture, il faut prendre en considération la manière dont les informations ont été stockées. En effet, la réplication uniforme risque d'aboutir à une duplication des réponses lors d'un processus de lecture. La solution de lecture doit donc extraire efficacement les informations tout en réduisant le taux de duplication de messages. Le délai et le taux de succès d'accès à l'information sont les caractéristiques essentielles de la qualité de service du protocole à inventer.

Pour terminer, les réseaux de capteurs dans le contexte de notre travail se distinguent par i) une densité importante des nœuds ii) des micro-nœuds avec ressources limitées (en particulier la ressource d'énergie), iii) l'impossibilité de changer les batteries/nœuds puisqu'ils sont enfouis dans le béton. Aussi une de nos contraintes est que les protocoles soient simples en limitant notamment le nombre de messages échangés entre les nœuds du réseau de capteurs et cela pour maximiser sa durée de vie.

V. Conclusion

Dans ce chapitre, le paradigme de matière communicante a été présenté. Son intérêt dans la gestion du cycle de vie des produits a été illustré à travers plusieurs champs applicatifs. Le but de cette thèse est de gérer les informations dans un réseau de capteurs sans fil enfoui dans le béton. Dans la suite de ce mémoire, nous allons nous attacher à étudier les solutions existantes dans la littérature pour stocker et lire des données dans un réseau de capteurs afin de développer les modes de communication les plus appropriés à notre contexte de travail. C'est dans ce sens que nous proposons dans le chapitre 2 un état de l'art sur les protocoles de stockage et les protocoles de lecture dans les réseaux de capteurs sans fil.

CHAPITRE II

Etat de l'art sur les protocoles de stockage et de lecture dans les réseaux de capteurs sans fil

I. Introduction

Puisque notre objectif est de proposer des protocoles de stockage et de lecture d'information dans un préfabriqué en béton embarquant un réseau de capteurs sans fil, il est nécessaire de comprendre les principes et fondements de la distribution des informations au sein de ce type de réseau. C'est en ce sens que nous proposons tout d'abord dans ce chapitre, un état de l'art sur les réseaux de capteurs, ses contraintes, ses spécificités et ses architectures de communication. Ensuite, les recherches et les propositions menées sur les protocoles de dissémination/stockage et protocoles de lecture d'informations sont exposées. À la fin de ces deux parties, une synthèse est présentée afin d'identifier les meilleurs mécanismes et modes de communication pour notre contexte applicatif.

II. Les réseaux de capteurs sans fil

Les réseaux de capteurs sans fil sont souvent composés d'un nombre important de nœuds. Ces nœuds sont des entités capables d'opérer en toute autonomie afin de collecter, traiter et envoyer des données relatives à leur environnement. Déployés de façon à couvrir un territoire donné, les capteurs communiquent par voie radio afin de concentrer l'information sur une ou plusieurs stations collectrices situées au cœur ou en bordure du territoire, appelées nœuds puits. Ces entités doivent être équipées d'une batterie qui est une source d'énergie non durable (sauf si le nœud est équipé par un module de récupération d'énergie), ce qui représente la contrainte la plus gênante pour la survie de tels réseaux.

Les réseaux de capteurs ne définissent pas une architecture préexistante puisque les nœuds peuvent être statiques ou mobiles, formant ainsi un réseau sans infrastructure prédéfinie (réseau Ad Hoc).

II.1. Architecture des réseaux de capteurs

Les nœuds de capteurs forment un réseau sans fil Ad Hoc (voir figure 14). Chaque nœud a pour mission la collection des données et leur routage vers le nœud puits. A son tour, le nœud puits transmet les informations reçues à l'utilisateur final à travers un réseau local (donc passerelle vers l'IoT), l'Internet ou un satellite.

Les topologies dans les réseaux de capteurs dépendent des applications et des techniques utilisées pour faire acheminer l'information des capteurs au nœud puits (Guo and Zhang, 2014). Le processus d'acheminement de l'information des capteurs au nœud puits peut prendre quatre formes. Dans les topologies à plat, les capteurs peuvent communiquer directement avec le puits en utilisant une forte puissance de transmission radio (figure 15(a)) ou via un mode multi-sauts avec des puissances très faibles (figure 15(b)) alors que dans les topologies hiérarchisées, le nœud chef du cluster transmet directement les données au puits (figure 15(c)) ou via un mode multi-sauts entre les chefs des clusters (figure 15(d)).

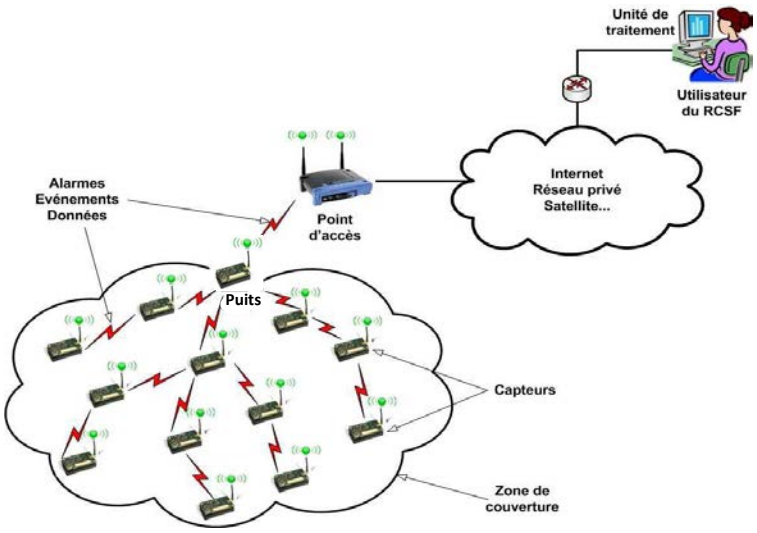


Figure 14. Architecture d'un réseau de capteurs sans fil

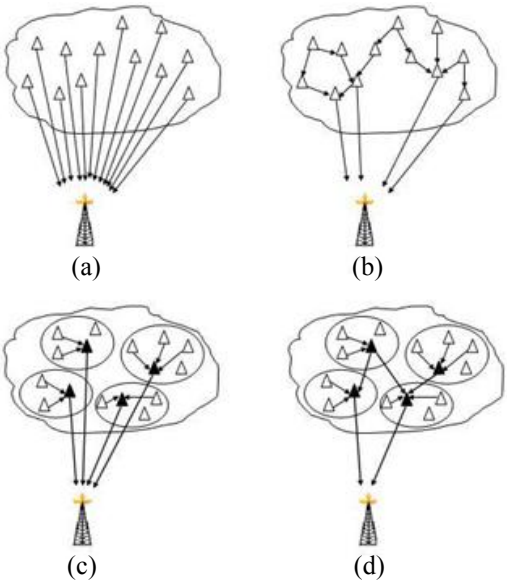


Figure 15. Types de communication dans un réseau de capteurs sans fil

II.2. Caractéristiques des réseaux de capteurs

Les principales caractéristiques des réseaux de capteurs se résument comme suit (Rashid and Rehmani, 2016):

- Les réseaux de capteurs se composent généralement d'un nombre très important de nœuds pour garantir une couverture totale de la zone à surveiller.
- Les nœuds disposent d'une énergie limitée (par exemple une pile AA), certains tomberont donc en panne avant les autres.
- La topologie du réseau de capteurs change fréquemment à cause de trois facteurs essentiels :
  - La mobilité des nœuds : les nœuds de capteurs peuvent être attachés à des objets mobiles qui se déplacent librement et arbitrairement, introduisant ainsi une topologie instable du réseau.

- La défaillance des nœuds : comme l'autonomie énergétique des nœuds est limitée, la topologie du réseau n'est pas fixe car les nœuds « morts » sont simplement supprimés (sauf dans le cas où le nœud est équipé d'un module de récupération d'énergie).
- L'ajout de nouveaux nœuds : des nouveaux nœuds peuvent facilement être rajoutés.

II.3. Contraintes des réseaux de capteurs

Les principaux facteurs et contraintes influençant l'architecture des réseaux de capteurs peuvent être résumés comme suit :

- L'auto-organisation : l'auto-organisation est impérative pour un réseau de capteurs afin de garantir sa maintenance. Au vu des différentes raisons aboutissant à une topologie instable du réseau de capteurs, ce dernier devra être capable de s'auto-organiser pour continuer à fonctionner.
- La tolérance aux pannes : en cas de défaillance d'un ou de plusieurs nœuds, le réseau doit être capable de maintenir ses fonctionnalités sans interruption. Cette défaillance peut être causée par une perte d'énergie ou par une panne physique.
- L'environnement de déploiement : les nœuds capteurs doivent être conçus d'une manière à résister aux différentes et sévères conditions de l'environnement : forte chaleur, pluie, etc.
- La consommation d'énergie : un nœud de capteurs est limité en énergie. Dans la plupart des cas, le remplacement de la batterie est impossible ce qui veut dire que la durée de vie d'un nœud dépend de l'autonomie de sa batterie. Pour limiter ce problème, des systèmes de récupération d'énergie (Energy Harvesting System) peuvent être associés au nœud (Wan et al., 2011).

II.4. Architecture d'un nœud de capteurs

Un nœud de capteurs sans fil est un petit dispositif électronique capable de mesurer une valeur physique environnementale (température, lumière, pression, etc) et de la communiquer à un centre de contrôle via une station de base. Il est composé de quatre unités de base (voir figure 16) :

- L'unité d'acquisition : elle est généralement composée de deux sous-unités, les capteurs et le convertisseur Analogique-Numérique. Les capteurs obtiennent des mesures sur les paramètres environnementaux, ces mesures sont ensuite converties par un convertisseur Analogique-Numérique afin de pouvoir être traitées par l'unité de traitement.
- L'unité de traitement : elle est composée de deux interfaces, une interface avec l'unité d'acquisition et une autre avec le module de transmission. Elle contrôle les procédures permettant au nœud de collaborer avec les autres nœuds pour réaliser les tâches d'acquisition et de stockage des données collectées.

- Un module de communication (Emetteur/Récepteur) : il est responsable de toutes les communications via un support de communication radio qui relie le nœud au réseau.
- Batterie : elle alimente les unités et les modules du nœud.

Il existe des nœuds qui sont dotés d'autres composants additionnels : les systèmes de localisation tels que le GPS (Global Position System) et des systèmes lui permettant de se déplacer.

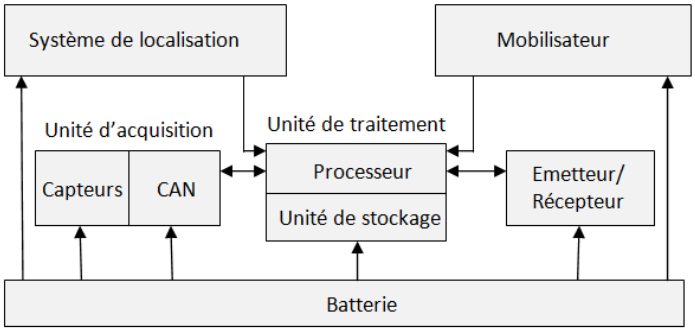


Figure 16. Architecture d'un nœud de capteurs sans fil

II.5. Modèles de communication

La figure 17 représente le modèle de communication souvent utilisé sur les réseaux de capteurs sans fil. Le rôle de ce modèle consiste à standardiser la communication entre les composants du réseau afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles. Ce modèle comprend cinq couches qui ont les mêmes fonctions que celles du modèle OSI ainsi que trois plans pour la gestion d'énergie, la gestion de la mobilité et la gestion des tâches. Le but d'un système en couches est de séparer le problème en différentes parties selon leur niveau d'abstraction. Chaque couche du modèle communique via des services avec une couche adjacente (celle du dessus ou celle du dessous).

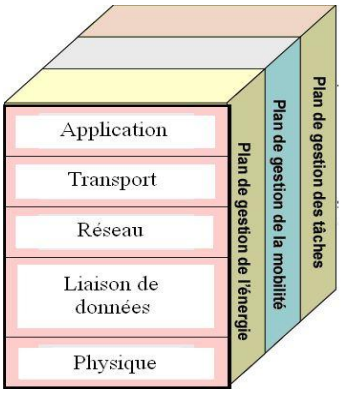


Figure 17. Pile protocolaire des réseaux de capteurs sans fil

II.5.1. La couche application

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

II.5.2. La couche transport

La couche transport aide à gérer le flux de données si le réseau de capteurs l'exige. Elle permet de segmenter les données issues de la couche application. A la réception, elle réordonne et rassemble les segments venus de la couche réseau avant de les envoyer à la couche application.

II.5.3. La couche réseau

La couche réseau achemine les données fournies par la couche transport. Son but principal est de trouver une route vers le puits en optimisant l'utilisation de l'énergie des nœuds.

II.5.4. La couche liaison de données

La couche liaison est responsable du multiplexage des flux des données, du contrôle d'accès au média et du contrôle des erreurs.

II.5.5. La couche physique

La couche physique est responsable du support radio acheminant les données envoyées entre les nœuds.

II.5.6. Plan de gestion d'énergie

Le plan de gestion d'énergie contrôle l'utilisation de la batterie. Par exemple, après la réception d'un message, le nœud éteint son récepteur afin d'éviter la duplication des messages déjà reçus. En outre, si le niveau d'énergie devient bas, le nœud diffuse à ses voisins une alerte les informant qu'il ne peut pas participer au routage. L'énergie restante est réservée au captage.

II.5.7. Plan de gestion de mobilité

Le plan de gestion de mobilité détecte et enregistre le mouvement du nœud. Ainsi, un nœud peut garder trace de ses nœuds voisins. En déterminant leurs voisins, les nœuds de capteurs peuvent balancer l'utilisation de leur énergie et la réalisation des tâches.

II.5.8. Plan de gestion de tâche

Le plan de gestion de tâche balance et ordonnance les différentes tâches de capture de données dans une région spécifique. Il n'est pas nécessaire que tous les nœuds de cette région effectuent la tâche de capture en même temps, certains nœuds exécutent cette tâche plus que d'autres selon leur niveau d'énergie.

II.6. Les modèles de transmission des données dans les réseaux de capteurs

La transmission des données dans les réseaux de capteurs peut se faire suivant plusieurs modèles dont on en distingue trois principaux (Singh et al., 2008): modèle orienté événement, modèle orienté requête et modèle à flux continu.

II.6.1. Le modèle orienté événement

La génération et la transmission des paquets des données sont commandées par l'apparition d'un événement environnemental. La plupart des applications orientées événements sont des applications critiques et temps réel. La réussite de ces applications repose essentiellement sur la détection de l'événement et la rapidité des prises de réaction nécessaires pour assurer l'aspect temps réel. L'inconvénient majeur de ce modèle est la redondance des données. En fait, les nœuds excités par le même évènement envoient la même information au nœud puits.

II.6.2. Le modèle orienté requête

La collecte des informations sur l'état de l'environnement par le modèle orienté requête est initiée par des interrogations envoyées par le nœud puits. La plupart des applications orientées requête sont des applications interactives et leur tolérance aux délais dépend de l'urgence de l'interrogation. Le modèle orienté requêtes peut être utilisé pour contrôler et configurer les nœuds. Seul le nœud jouant le rôle de puits peut émettre des interrogations ou des commandes.

II.6.3. Le modèle à flux continu

Dans le modèle à flux continu, les nœuds de capteurs envoient les informations d'une manière continue au nœud puits suivant un volume de trafic prédéterminé.

III. Protocoles de stockage d'information dans les réseaux de capteurs

Les protocoles de stockage et réplication des données dans les réseaux de capteurs sans fil ont été développés principalement dans la littérature pour la gestion des nœuds puits mobiles (Tunca et al., 2014). Il existe deux approches principales : *réactive* et *proactive*. Dans la première, les nœuds réagissent à un événement en diffusant les données vers les nœuds qui sont situés à proximité des positions de l'événement. Dans l'approche proactive, les nœuds distribuent leurs données vers tous les noeuds ou un sous-ensemble de nœuds qui ont le rôle d'unité de stockage, en prévision des futurs événements. L'approche proactive est divisée en stratégies *structurées* et *non structurées* comme illustré sur la figure 18. Dans la première stratégie, les nœuds de stockage forment généralement une structure virtuelle (p.ex. grille, ligne, rail, etc.) dans le réseau de capteurs, les données pouvant être ensuite récupérées en envoyant une requête de lecture vers la structure de stockage. Cependant, pour la stratégie non structurée, les données sont stockées dans tout le réseau. Dans ce dernier cas, la façon de

distribuer les données dans le réseau détermine si l'information est stockée uniformément ou non. Dans la suite, les différentes techniques sont présentées pour chaque classe.

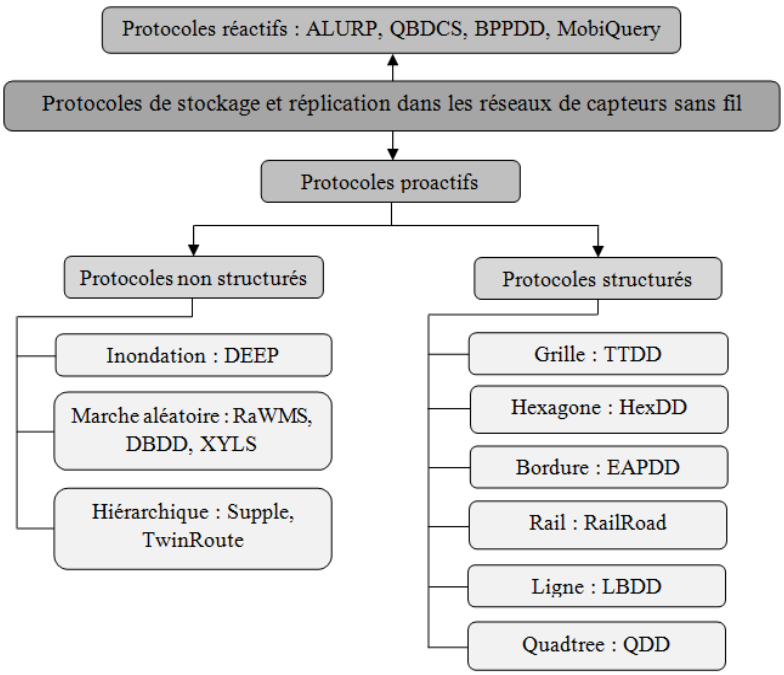


Figure 18. Les protocoles de stockage de données dans les réseaux de capteurs sans fil

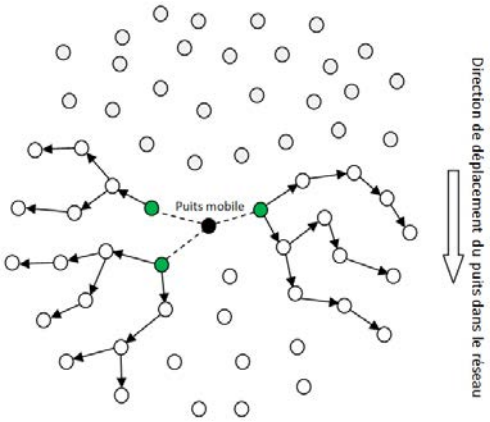
III.1. Protocoles de dissémination réactifs

a) ALURP : ALURP (Adaptive Local Update-based Routing Protocol) (Wang et al., 2009) est un protocole de réplication et collection réactif. Avec ALURP, le puits mobile inonde le réseau par un message qui contient les informations relatives à son emplacement géographique. Donc, les nœuds du réseau peuvent envoyer leurs données au puits par un routage géographique. Une différence principale entre ALURP et un protocole de routage géographique traditionnel est que ce dernier transmet les données directement au puits saut par saut. Par contre en utilisant ALURP, la source réplique ses données vers l'ensemble des nœuds situés dans la zone de localisation de puits (zone de destination). Le puits est localisé dans une zone nommée Virtual Center (*VC*). *S* qui est le nœud source transmet le paquet de données aux nœuds dans la zone centrée sur *VC*. Quand tous les nœuds sources ont distribué leurs données vers la zone destinataire, le puits peut ensuite diffuser sa requête dans sa zone de localisation locale pour collecter les informations. Lorsque le puits se déplace hors de sa zone, il inonde sa nouvelle localisation dans le réseau. La nouvelle zone de destination sera construite et le processus de routage et de réplication par les sources se répètent de la même manière.

b) QBDCS: QBDCS (Query-Based Data Collection Scheme) (Cheng et al., 2009) s'appuie sur la mobilité prévisible du puits. Lors du passage de ce dernier à proximité du réseau, il diffuse la requête vers la zone d'intérêt dans le réseau par un routage géographique. La

requête contient la position, la vitesse et la direction du puits. Lors de la réception de la requête, les nœuds source dans la zone d'intérêt estiment la future position du puits. Le paquet de réponse sera acheminé vers la zone de position estimée. Chaque nœud de cette zone vérifie si le puits est déjà passé ou s'il n'est pas encore dans sa portée radio. Si le puits n'est pas passé, le nœud stocke la donnée et attend le passage du puits dans son champ de transmission radio pour lui transmettre les données. Si le puits est déjà passé, le nœud stocke la donnée pendant un temps d'attente $T_{deadline}$ jusqu'à ce qu'il détecte un autre passage de puits. En cas de dépassement de $T_{deadline}$, le paquet est supprimé.

c) MobiQuery : MobiQuery (Lu et al., 2005) est un algorithme de réplication et collection réactif qui réagit au mouvement du puits en temps réel. MobiQuery s'exécute en cinq étapes. Au début, quand un ensemble de nœuds détectent la présence d'un puits mobile, ils lancent un processus de construction d'un arbre de routage comme illustré sur la figure 19. Dans une deuxième étape, chaque nœud dans le réseau traite un seul message de construction d'arbre lié à un nœud de voisinage du puits. Ce processus est limité de telle sorte qu'aucun arbre dépasse une profondeur déterminé TD_{hops}. De plus, si un nœud a récemment participé à un autre arbre de routage de donnée pour une position antérieure du puits, il ne participe à aucun nouvel arbre jusqu'à l'expiration d'une période de temps nommée *justCoveredTimer*. Le but de la première restriction est de limiter la profondeur de chaque arbre et donc de limiter l'énergie consommée pour leur construction et lors de l'acheminement de données. La deuxième restriction suit l'intuition qu'il n'est pas utile d'élargir l'arbre vers une région où il y a peu de nouvelles données à signaler au puits. Enfin, lors de la troisième étape, les nœuds de chaque arbre transmettent leurs données vers leurs parents dans l'arborescence. Ces données sont stockées et agrégées au niveau de la racine de l'arbre pour les transmettre ensuite au puits quand celui-ci est dans sa portée radio. L'arbre est détruit automatiquement après un court intervalle de temps lorsque le puits est hors de portée radio de la racine de l'arbre (nœud de stockage).



Nœud qui a participé à la construction de l'arbre de routage pour une position antérieure du puits

Nœud qui a détecté le passage du puits

Figure 19. Dissémination réactive par MobiQuery

d) BPPDD : BPPDD (Balancing Push and Pull Data Dissemination) (Liu et al., 2007) propose une approche hybride *push-pull* pour la réplication et la récupération des données dans une distribution d'une grille des nœuds de capteurs. Lors de la détection d'un événement dans le réseau (comme l'apparition d'un puits mobile), les nœuds transmettent leurs données (*push*) au-dessus et au-dessous de son emplacement pour construire une ligne de réplication verticale de longueur *L*. La requête de collection est ensuite transmise horizontalement par le puits mobile. Ce processus est exécuté en se basant sur les coordonnées et les positions géographiques afin de sélectionner les nœuds dans le sens vertical et horizontal pour les données et la requête, respectivement. Pour illustrer ce principe, on peut prendre un réseau avec une distribution de n^2 nœuds, chaque nœud étant à la position (i,j) avec $0 \leq i, j < n$. On suppose que chaque nœud peut communiquer uniquement avec quatre nœuds voisins les plus proches. Pour un nœud à la position (i,j), les nœuds de réplication de données sont localisés aux (i,j+1), (i,j+2),…,(i,j+L/2) et (i,j-1), (i,j-2),…,(i,j-L/2+1).

III.2. Protocoles de dissémination proactifs

III.2.1. Protocoles structurés

Dans les approches proactives structurées, les nœuds forment généralement une structure de stockage virtuelle (grille, ligne, rail, etc.) dans le réseau de capteurs comme illustré par la figure 20. Chaque technique est décrite dans cette section.

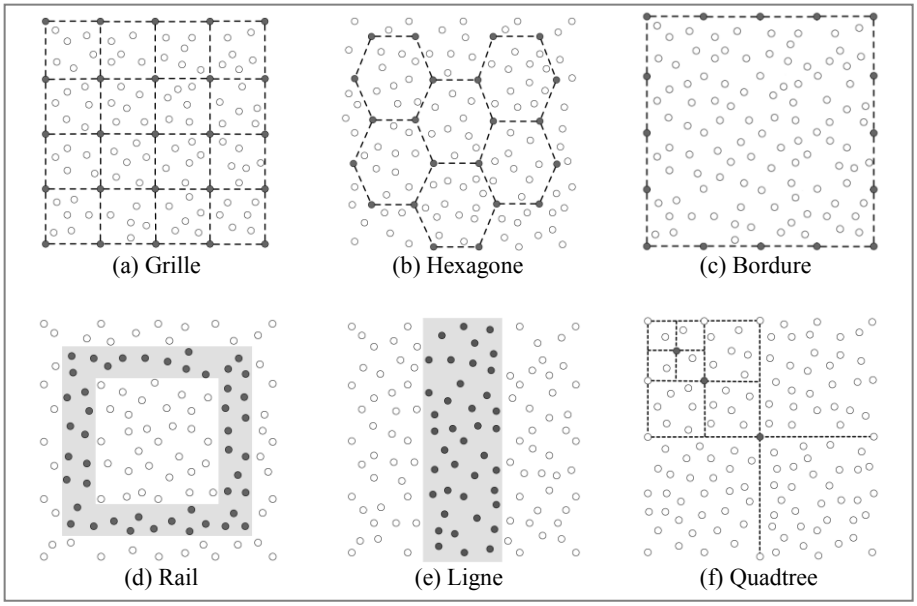


Figure 20. Stratégies de stockage proactif structuré dans les réseaux de capteurs sans fil

a) TTDD : TTDD (Two-Tier Data Dissemination) (Luo et al., 2005) est une approche de dissémination de données proposée pour résoudre le problème de puits mobiles multiples. Au lieu de propager une requête de collection à partir de chaque puits vers tous les nœuds de réseau, TTDD utilise une structure de grille de sorte que seuls les nœuds situés à proximité des puits répondent aux requêtes. Lors de la capture d'une information et au lieu d'attendre

passivement les requêtes provenant de puits, la source construit une structure de grille virtuelle à travers le réseau comme l'illustre la figure 20(a). Puis, elle réplique ses données en chaque point de la grille en se basant sur un algorithme de calcul des positions géographiques des nœuds voisins. Pour illustrer l'algorithme, on considère un champ de réseau à deux dimensions. Une source à l'emplacement L_s=(x, y) divise le champ en une grille de cellules. Chaque cellule est un carré de dimension $\alpha \times \alpha$. Les points de réplication de la grille sont situés à L_p=(x_i, y_j), de telle sorte que $\{x_i = x + i\alpha, y_j = y + j\alpha; i, j = \pm 0, \pm 1, \pm 2, \ldots\}$.

La source calcule les positions géographiques L_p de ses quatre voisins de grille (points de réplication) en utilisant la position (x, y) et la taille de cellule α. Ensuite, elle transmet la donnée aux nœuds voisins qui sont les plus proches des positions L_p à l'aide d'un acheminement géographique. Les nœuds de stockage continuent ce processus jusqu'à la fin de la réplication des données dans toute la grille.

b) HexDD : HexDD (Hexagonal cell-based Data Dissemination) (Erman et al., 2012) crée une infrastructure virtuelle basée sur une structure hexagonale par localisation et routage géographique comme indiqué sur la figure 20(b). Les frontières de l'hexagone dans le réseau sont des lignes de réplication de données ainsi que des lignes de rendez-vous avec les requêtes. Au lieu d'acheminer les données à travers les nœuds au milieu des cellules, les données sont transmises sur les lignes de l'hexagone à l'aide d'un routage géographique. L'objectif est de stocker ces données dans les nœuds sur les lignes frontières de sorte que les puits mobiles puissent facilement les recueillir en utilisant le même mécanisme d'acheminement pour les requêtes. HexDD est une approche purement géographique (pas d'inondation pour la configuration de la route), la seule information requise est la position des nœuds. Quand la requête arrive à un nœud de frontière qui stocke l'information demandée, les données sont envoyées vers le puits à travers le chemin inverse parcouru par la requête.

c) EAPDD : EAPDD (Edge Agregation Point for Data Dissemination) (Doss et al., 2010) est un algorithme de dissémination qui stocke les informations générées par l'ensemble des nœuds du réseau dans des points d'agrégation situés sur la bordure de ce réseau comme illustré sur la figure 20(c). EAPDD est développé pour augmenter la durée de vie du réseau et pour réduire le temps de traitement des requêtes de collection. L'objectif d'EAPDD est d'organiser le réseau de telle sorte qu'il puisse clairement identifier le périmètre du réseau et sélectionner des nœuds d'agrégation et de stockage des données. Pour ce faire, EAPDD présente un algorithme distribué simple. Chaque nœud a son emplacement géographique. Tous les nœuds qui ont une distance inférieure à un seuil d par rapport à la bordure du réseau, s'auto-désignent pour être un nœud de bord. L'algorithme sélectionne un ensemble parmi les nœuds de bord, pour être des points d'agrégation de données nommé *AP* (Aggregation Point). Une fois identifié, chaque *AP* diffuse un message *Hello*. Ce message est transmis vers les

nœuds de bord voisin et s'arrête à un nœud qui a déjà reçu au moins un message *Hello* d'un autre *AP*. Grâce à ce processus, chaque nœud du réseau connait son *AP* le plus proche pour des opérations de stockage et de collection de données.

d) RailRoad : Railroad (Shin et al., 2005) adopte une infrastructure virtuelle appelée *rail* placée au milieu de la zone de déploiement du réseau de capteurs comme l'illustre la figure 20(d). Les nœuds à l'intérieur de cette structure sont appelés *nœuds de rail*. Chaque nœud doit connaître la distance qui le sépare du centre du réseau et la distance du nœud de bordure de réseau le plus proche (les distances sont calculées par le nombre de sauts). Chaque nœud peut déterminer localement s'il appartient au *rail* ou non en se basant sur ces valeurs. Les nœuds qui ne font pas partie du rail, mémorisent l'identifiant du nœud voisin qui mène vers le *rail*. En se basant sur l'emplacement du centre du réseau, chaque nœud de *rail* détermine dans quelle direction il se trouve dans cette structure. Lorsqu'une source génère des données, le message est transmis aux *nœuds rail* les plus proches. Pour récupérer les données, le puits envoie une requête en direction du centre du réseau. La requête se propage d'un *nœud rail* à un autre pour collecter et agréger les données demandées avant de les transmettre vers le puits.

e) LBDD : LBDD (Line-Based Data Dissemination) (Hamida and Chelius, 2008) définit une ligne verticale virtuelle de largeur *w* qui divise le réseau de capteurs en deux champs égaux, comme l'illustre la figure 20(e). La ligne passe par le centre du réseau de sorte que chaque nœud puisse facilement y accéder par communication horizontale. Les nœuds dans cette ligne sont appelés *inline*, tandis que les autres nœuds sont appelés *ordinary*. Cette ligne représente une région de rendez-vous pour le stockage des données et la requête de collection. La conception du LBDD suppose que chaque nœud connait sa position géographique, ainsi que les coordonnées géographiques de bordure du réseau. En se basant sur la position du nœud par rapport aux bordures du réseau et au paramètre *w*, le nœud décide facilement s'il est *inline* ou non. Par suite, quand un nœud *ordinary* génère de nouvelles données, il transmet le message vers le nœud *inline* le plus proche. Un puits mobile peut ensuite récupérer ces données en envoyant une requête dans une direction perpendiculaire à la ligne de stockage. La requête se propage à travers les nœuds *inline* jusqu'à atteindre le nœud qui stocke l'information demandée. Ce dernier transmet ensuite les données directement au puits.

f) QDD : QDD (Quadtree-based Data Dissemination) (Mir and Ko, 2006) définit une structure de réplication basée sur une technique de localisation nommée *Quadtree* (Cimen et al., 2003) qui consiste à partitionner l'espace de déploiement du réseau en différent quadrants successifs comme le montre la figure 20(f). Dans cette approche, un nœud source de données calcule un ensemble de points de rendez-vous en divisant successivement le champ de réseau en quatre quadrants logiques comme on peut le voir sur la figure 21, et les données sont envoyées aux nœuds qui sont les plus proches du centre de gravité de chaque quadrant. Le

routage géographique est utilisé pour transmettre les données vers les points de réplication. Le puits suit la même stratégie pour la collecte et la transmission de la requête.

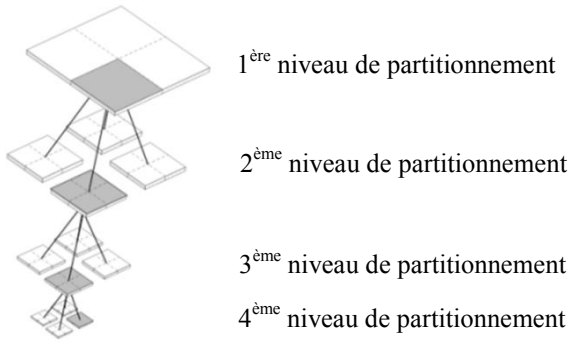


Figure 21. Partitionnement de réseau en quatre quadrants logiques successifs

III.2.2. Protocoles non structurés

a) DEEP : DEEP (Density-based proactivE data dissEmination Protocol) (Vecchio et al., 2010) est un protocole de dissémination proactif basé sur la densité de déploiement de réseau de capteurs sans fil. Son but est d'obtenir une distribution uniforme et efficace des données avec un taux de communication réduite. Il permet au puits collecteur mobile de recueillir une image représentative de toutes les informations captées dans le réseau en visitant un ensemble de x nœuds sur un total de n nœuds, où $x<<n$. DEEP combine l'algorithme d'inondations probabiliste RAPID (Drabkin et al., 2007) avec une solution de stockage probabiliste : lors de la première réception d'un message m, le nœud i stocke m avec une probabilité ps qui dépend de la taille souhaitée de la vue partielle (la vue d'un nœud sur un ensemble des informations générées par d'autre nœuds dans le réseau). La vue partielle correspond à la quantité d'informations stockées dans un nœud, provenant d'autres nœuds les ayant disséminées dans le réseau. À titre d'exemple, pour une vue partielle de taille $\sqrt{n}$, un nœud i stocke m avec une probabilité $ps = \frac{\sqrt{n}}{n}$. Ensuite, i rediffuse m dans son voisinage avec une probabilité $pr=min\left(1, \frac{\beta}{|N(i)|}\right)$, où $N(i)$ est le nombre des nœuds voisins de i (densité de voisinage) et β est le nombre moyen souhaité de retransmissions dans chaque voisinage.

b) RaWMS : RaWMS (RAndom Walk based Membership Service) (Bar-Yossef et al., 2008) est un algorithme de réplication uniforme aléatoire pour les réseaux de capteurs sans fil. RaWMS est basé sur la technique de marche aléatoire appelée Random Walk (RW). Chaque nœud source commence à plusieurs reprises un *RW*, dont le message porte un champ associé *Time-To-Live* (*TTL*) qui définit la longueur de la *RW*. Chaque *RW* traverse le réseau pour un nombre prédéfini de sauts. Le dernier nœud de chaque *RW* est responsable du stockage des données comme l'illustre la figure 22. Les auteurs ont prouvé que quand tous les *RW* sont terminés, chaque nœud a un ensemble de données provenant de différentes sources et une vue uniforme du réseau. De même que DEEP, cet algorithme permet à un puits collecteur mobile

de recueillir une image représentative de toutes les informations générées dans le réseau en visitant un ensemble de *x* nœuds << *n*.

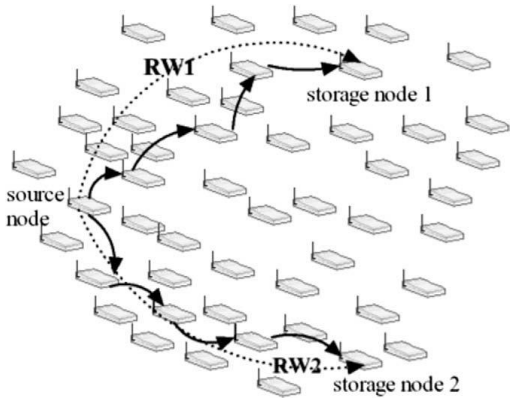


Figure 22. Réplication d'information par marche aléatoire (Bar-Yossef et al., 2008)

c) DBDD : DBDD (Double-Blind Data Discovery) (Shi et al., 2012) est un protocole de dissémination pour les réseaux de capteurs sans fil à grande échelle avec puits mobiles, où le puits et les sources de données ne connaissent pas a priori les emplacements des uns et des autres. DBDD adopte l'approche de la marche à ligne aléatoire *Random Line Walk* (*RLW*). Dans *RLW*, lorsqu'un nœud a une donnée à disséminer, il la transmet vers une direction prise au hasard pour un nombre de sauts successifs formant ainsi une ligne dite « aléatoire ». Pour que ce mécanisme fonctionne, l'information de direction relative des nœuds voisins est suffisante. Cette information peut être obtenue par une règle *cosinus* (Shi et al., 2012) dans le cas où les distances entre les voisins sont connues. *RLW* assure qu'un paquet de données se propage le long d'une ligne droite. Toutefois, il ne peut pas garantir que deux chemins de deux paquets de données (générés par deux nœuds différents) aient un angle prédéterminé entre eux, ce qui nécessite d'avoir la position des nœuds de capteurs. DBDD étend l'idée de *RLW* en exploitant la propriété géométrique du plan du réseau. Chaque nœud envoie ses données le long de deux lignes orthogonales qui correspondent aux quatre directions. Les requêtes émises par le puits sont envoyées dans les quatre directions de la même manière. En raison de la propriété géométrique orthogonale, les données et les requêtes ont une grande probabilité d'intersections. La transmission de données vers le puits se fait en prenant le chemin inverse de la requête.

d) XYLS : XYLS (Das et al., 2005) est un algorithme de réplication et collection proactive (dans un réseau de capteurs mobile) qui diffuse les données et les requêtes dans des directions orthogonales pour qu'une requête puisse croiser les données par la suite. Pour chaque nœud source de données, les nœuds qui sont situés le long de la direction nord/sud de sa zone géographique constituent pour lui des nœuds de réplication (réplication dans le sens vertical). De même, pour un nœud collecteur, les nœuds qui sont situés le long de la direction est/ouest forment les nœuds d'acheminement de requête (recherche dans le sens horizontal). De cette

manière chaque requête peut croiser l'un des nœuds qui stocke les données. XYLS a été proposé pour un réseau mobile. Lors du stockage des données, chaque nœud sélectionne parmi son voisinage les nœuds situés au nord et au sud, puis il envoie les données à ses voisins. Les données sont renvoyées après chaque déplacement du nœud à une distance *d* (seuil de déplacement au-dessus duquel les données sont répliquées). Pour gérer le stockage de données aux nœuds de réplication, chaque donnée dispose d'un délai de temporisation qui précise la validité de la donnée dans la mémoire d'un nœud. Le délai de temporisation est prévu sur la base de la mobilité du nœud source et la distance seuil *d*. Lors d'expiration de temporisateur, les données sont supprimées de la mémoire des nœuds de capteurs.

e) Supple : Supple (Viana et al., 2010) est une approche basée sur une structure hiérarchique (arbre de dissémination). C'est un protocole de réplication probabiliste pour les réseaux de capteurs sans fil. Supple comporte trois phases : la construction de l'arbre, la répartition des poids et la réplication des données. La première phase est une construction de l'arbre initiée par un nœud central (nœud racine) du réseau. Le nœud racine est chargé de recevoir et de reproduire les données capturées dans le réseau. La deuxième phase attribue un poid W_i à chaque nœud *i* qui représente la probabilité de stockage. Supple utilise le nombre de sauts d'un nœud *i* par rapport à la racine pour calculer W_i. Dans la dernière phase, les nœuds de capteurs envoient leurs données à la racine qui va ensuite transmettre plusieurs fois chaque donnée (*r* fois) tout au long des branches de l'arbre hiérarchique. Chaque nœud *i* de l'arbre stocke les données avec la probabilité W_i. Les auteurs affirment qu'un puits mobile visitant un petit ensemble de nœuds (environ $2{,}3\times\sqrt{n}$, *n* étant le nombre total de nœuds du réseau) peut récupérer toutes les données générées dans le réseau.

Cependant, Supple ne considère pas les problèmes liés à la consommation d'énergie et à la grande charge de communication au niveau des nœuds les plus proches de la racine. Pour faire face à ce problème, un protocole de stockage de données distribuées pour les réseaux de capteurs hétérogènes appelé ProFlex est présenté dans (Maria et al., 2013). ProFlex nomme certains nœuds du réseau *H-capteur* (nœuds puissants). Un *H-capteur* est un nœud avec des capacités améliorées de calcul, de stockage, de batterie et de puissance de communication. Plusieurs arbres (structures de réplication) sont construits selon le nombre et le positionnement des nœuds *H-capteur* (un *H-capteur* est le centre de chaque sous-arbre). Ainsi, la charge de communication n'est plus concentrée autour d'un seul nœud racine, mais distribuée sur plusieurs racines *H-capteur* puissants.

f) TwinRoute : TwinRoute (Wohlers et al., 2009) est un algorithme qui construit et entretient des arbres de dissémination de données avec des racines prédéfinis comme nœuds de stockage. Il suppose généralement des trajectoires fixes ou aléatoires de puits sur lesquelles les nœuds de stockage sont sélectionnés. En gardant les traces des visites de puits, il est possible d'identifier les nœuds qui sont les plus visités et susceptibles de fournir des données à

un puits afin de les utiliser comme nœuds de stockage. TwinRoute sélectionne les nœuds de stockage en se basant sur l'historique des visites de puits, et construit des arbres de dissémination dont ils sont la racine. Cela permet ainsi aux nœuds restant de transmettre de manière proactive leurs données vers le nœud racine de stockage le plus proche. Plus précisément, TwinRoute comporte quatre tâches. La première tâche consiste à garder la trace des visites de puits. Quand un puits est détecté dans le champ de transmission radio, le nœud met à jour sa liste de visites de puits. La deuxième tâche implique un schéma de sélection de nœuds de stockage. Un délai de stockage *SDT* est défini, qui représente le temps maximal d'attente des paquets de données aux nœuds de stockage avant d'être livrés aux puits mobiles. En se basant sur les informations des visites récentes de puits, si un nœud estime livrer la majorité de ses paquets en un délai $D<SDT$, le nœud devient alors un nœud de stockage et diffuse un message de construction d'arbres. La troisième tâche décrit comment un nœud de réseau traite le message de construction d'arbres de sorte qu'il s'associe à un chemin menant à son plus proche nœud de stockage (en terme de nombre de sauts). Finalement, les nœuds faisant partie de chaque arbre transmettent leurs données le long des routes établies vers les racines (les nœuds de stockage). Un nœud de stockage envoie directement les données au puits quand ce dernier est dans son champ de transmission radio.

III.3. Analyse et synthèse

Le tableau 1 résume les techniques utilisées par chaque protocole de dissémination présenté dans cette section. Pour mémoire, l'objectif de cet état de l'art a été d'identifier des protocoles applicables à notre cas d'étude, c'est-à-dire ceux respectant les contraintes posées par notre positionnement scientifique, décrit dans le chapitre précédent, qui sont :

- Dissémination de données uniforme,
- Densité fonction de l'importance de l'information,
- Forte densité de nœuds (par exemple 1000 nœuds) avec des ressources très limitées.

Ce tableau présente donc les travaux selon trois principales : protocole réactif ou non, caractéristiques du stockage utilisé et caractéristiques de la réplication.

De cette étude, nous avons pu tirer 6 protocoles proactifs (2 structurés et 4 non structurés) capables d'assurer une réplication uniforme d'une information dans un réseau de capteurs : TTDD, HexDD, DEEP, RaWMS, Supple et ProFlex. Comme nous l'avons souligné dans le premier chapitre, le réseau de capteurs intégré dans le préfabriqué en béton possède certaines spécificités : forte densité de nœuds, nœuds de tailles réduites (micro-nœuds) avec des ressources très limitées (mémoire, énergie, capacités de calcul, etc.), remplacement impossible des nœuds et des batteries. L'utilisation de micro-nœuds pauvres en ressource et enfouis dans le béton rend l'utilisation des techniques de localisation géographique difficile (Tran-Dang et al., 2014). Donc, l'utilisation des techniques de réplication sur une grille

TTDD ou hexagone HexDD compliqués à appliquer. De plus, les nœuds dans le préfabriqué étant identiques (pas d'utilisation des nœuds plus puissants que les autres), l'application de technique à nœuds hétérogènes tel que ProFlex n'est pas envisageable.

DEEP, RaWMS et Supple s'appuient sur des stratégies de réplication uniforme se basant sur les paramètres physiques du nœud de capteurs (p.ex. la mémoire et l'énergie), la topologie du réseau et la connaissance de son voisinage pour répliquer les données à travers le réseau de capteurs. En ce sens, ils sont tous trois de bons candidats pour l'implémentation de notre protocole d'écriture, et constituent nos protocoles de référence. Il est cependant à noter qu'aucun de ces protocoles ne considèrent les propriétés intrinsèques des données diffusées telle que, dans notre cas, l'importance de l'information. Par conséquent, les solutions existantes ne permettent pas de faire varier la densité de réplication d'une information suivant son niveau d'importance tout en gardant une réplication uniforme dans tout le réseau quelle que soit la densité. L'apport d'une solution à ce problème constitue la contribution principale de notre travail dans la phase de dissémination et de stockage.

Tableau 1. Résumé des protocoles de stockage d'information dans les réseaux de capteurs sans fil

Protocole de stockage	Réactif	Proactif		Caractéristique du stockage utilisé			Caractéristiques de la réplication				
		Structuré	Non-structuré	Probabiliste	Comptage de sauts	Déterministe	Uniformité de réplication	Utilisation de la localisation géographique	Utilisation de la distance inter-nœuds	Hiérarchique	Nœuds hétérogènes
ALURP	✓					✓		✓			
QBDCS	✓					✓		✓			
BPPDD	✓					✓		✓			
MobiQuery	✓					✓				✓	
TTDD		✓				✓	✓	✓			
HexDD		✓				✓	✓	✓			
EAPDD		✓				✓		✓			
RailRoad		✓				✓		✓	✓		
LBDD		✓				✓		✓			
QDD		✓				✓		✓	✓		
DEEP			✓	✓			✓				
RaWMS			✓		✓		✓				
DBDD			✓		✓				✓		
XYLS			✓			✓		✓			
Supple			✓	✓			✓			✓	
ProFlex			✓	✓			✓			✓	✓
TwinRoute			✓			✓				✓	

Pour concevoir notre solution, il est souhaitable d'utiliser les meilleures caractéristiques des protocoles sélectionnés. La dissémination de données d'un nœud à un autre peut s'appuyer sur une structure hiérarchique de routage comme celle employée par Supple, ou sur des

mécanismes d'inondation probabiliste utilisés par DEEP, ou aussi sur le mécanisme de marche aléatoire de RaWMS. L'application d'une structure hiérarchique dans le contexte de la matière communicante semble être délicate. En effet, ce type de structuration nécessite des phases de synchronisation et de construction entre les nœuds dans la matière, en complément de phases de mise à jour nécessaires pour maintenir la structure. Cela peut aboutir à une surcharge de communication supplémentaire dans le processus de dissémination. Dans nos travaux, un premier choix a été de ne pas considérer de structure hiérarchique dans le réseau de capteurs.

Le mécanisme d'inondation garantit un taux de réception du message émis plus grand que le mécanisme de marche aléatoire (Vecchio et al., 2010). En effet, l'inondation garantit en une seule diffusion la réception du message par chaque nœud du réseau, à contrario de la marche aléatoire où des nœuds peuvent ne pas recevoir de message. Pour que le taux de réception soit élevé en utilisant la marche aléatoire, plusieurs exécutions (des centaines de fois) de cette technique sont donc nécessaires ce qui réduit la durée de vie des nœuds. Pour cette raison, nous avons choisi l'inondation comme mécanisme pour la dissémination de données.

Cependant, l'inondation de données dans des réseaux à micro-nœuds très limités en ressources doit être judicieuse. La quantité de données transmises sur le réseau doit être réduite autant que possible afin de maximiser la durée de vie du réseau (en fait, la communication sans fil consomme plus d'énergie que toute autre activité du nœud de capteurs (Nechibvute et al., 2012)). Donc, tout mécanisme de diffusion des données pour ces micro-nœuds doit reposer sur un faible taux de communications définies par le nombre de transmissions d'un message dans le réseau pendant le processus de dissémination.

DEEP et Supple se basent sur le stockage probabiliste et RaWMS sur le stockage par compteur de sauts. Une étude comparative entre le stockage probabiliste et le stockage par compteur de nombre de sauts a été réalisée au cours d'un stage de master au CRAN (voir Annexe). Cette étude publiée dans (Mekki et al., 2014a) (Mekki et al. 2016a) a montré que le stockage probabiliste offre une réplication plus uniforme que le stockage par compteur de nombre de sauts. Nous avons déterminé par simulation que le stockage par comptage de nombre de sauts aboutit à des zones où la matière est vide d'information. Nous avons donc choisi un stockage probabiliste dans nos travaux. Cependant, un tel mécanisme pourrait conduire à une saturation de la mémoire si une probabilité élevée est utilisée (Mekki et al., 2016a). À titre d'exemple, si la probabilité est mise à 1, les données sont stockées dans tous les nœuds de la matière (surcharge de mémoire après quelques opérations de stockage). Notre protocole emploie donc un mécanisme contrôlant la réplication de données dans chaque voisinage. Une variable binaire *NS* a été définie dans l'entête du message. Si un nœud stocke l'information dans sa mémoire interne, *NS* est mis à 1 pour informer les nœuds voisins, d'où ces derniers ne stockent pas l'information. L'avantage apporté par ce mécanisme de

réplication contrôlé sera démontré dans le dernier chapitre de ce mémoire, en particulier, en terme d'augmentation de la capacité de stockage du préfabriqué.

Le protocole que nous proposons est nommé ***USEE*** : *Uniform Storage and Energy Efficient data dissemination protocol*, et il est présenté dans (Mekki et al., 2016b). Le protocole USEE est détaillé dans le chapitre suivant et évalué par simulation dans la dernière partie de ce rapport.

IV. Protocoles de lecture et recherche d'information dans les réseaux de capteurs

Les protocoles de recherche d'information dans les réseaux de capteurs sont de type orienté requêtes (voir section II.6.2 de ce chapitre). Le puits envoie une *requête* à un nœud ou à un groupe de nœuds distants pour obtenir les informations de leurs capteurs ou des informations stockées dans leurs mémoires. Les nœuds qui ont l'information demandée répondent par un message *réponse* qui contient ces données. Cet échange de message pour la recherche d'information est nommé *requête/réponse*.

Différents protocoles de recherche d'information ont été proposés dans la littérature pour les réseaux de capteurs sans fil. Ils peuvent être classés en protocoles à *deux phases requête/réponse* et protocoles à une *seule phase requête/réponse* comme l'illustre la figure 23. Le reste de cette section décrit le fonctionnement de chaque classe ainsi que les différents protocoles existants.

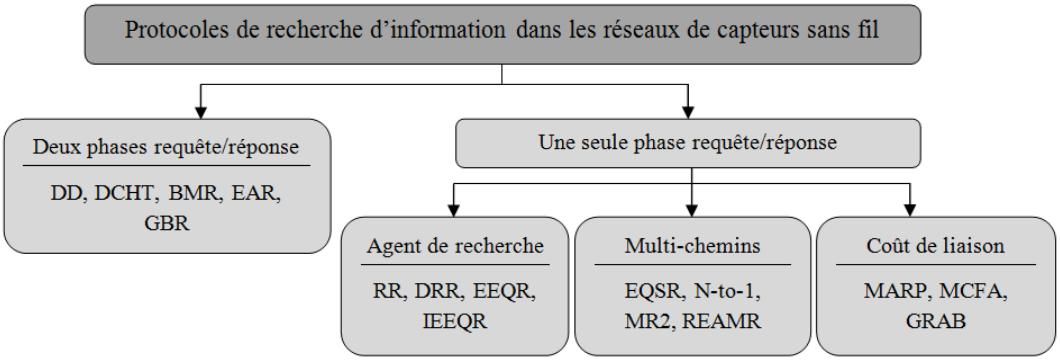


Figure 23. Les protocoles de recherche d'information dans les réseaux de capteurs sans fil

IV.1. Protocoles à deux phases requête/réponse

Dans la classe *deux phases requête/réponse*, le puits identifie les données demandées par un ensemble d'attributs. Ces attributs se propagent à travers le réseau dans un message requête. Les informations mises en cache à partir des recherches antérieures, comme l'information géographique des nœuds ou des filtres spécifiques à l'application, peuvent être utilisés pour optimiser la propagation de la requête. Cependant, sans ces informations, la requête doit inonder le réseau pour trouver les données recherchées.

Lorsqu'une requête arrive à un nœud source, ce dernier envoie un message de données d'exploitation des chemins menant au puits. Comme avec le message requête, soit ce transfert

est limité en utilisant des informations mises en cache, soit il est envoyé à tous les nœuds. Lorsque les données d'exploration atteignent le puits, ce dernier renforce son voisin préféré pour établir un chemin vers la source (en considérant par exemple un nœud voisin à faible latence, à haute qualité de liaison radio ou encore à haute énergie restante). Le voisin renforcé par le puits, à son tour, renforce son voisin vers la source. Cela continue jusqu'à la construction d'un chemin à haute qualité de service liant le puits à la source. Finalement, la source envoie ces données uniquement sur le chemin renforcé plutôt que vers tous les voisins. Dans ce qui suit, les principales techniques de recherche à deux phases requête/réponse sont présentées.

a) Directed Diffusion: Directed Diffusion (DD) (Intanagonwiwat et al., 2000) est l'un des premiers protocoles de recherche à deux phases requête/réponse. Il fonctionne comme suit : le nœud puits commence à diffuser, vers tous les nœuds, un message requête. Ce paquet sera acquitté par un autre message appelé *Gradient*. Ce dernier est une réponse de la part du voisin recevant la requête. En utilisant la requête et les gradients, plusieurs chemins peuvent être établis entre le puits et la source comme cela est illustré dans les figures 24(a) et 24(b). L'un de ces chemins est sélectionné par renforcement depuis le puits (voir la figure 24(c)). Si ce chemin échoue, un nouveau chemin doit être identifié. Finalement, la source commence la transmission de ses données à travers le chemin renforcé (figure 24(d)).

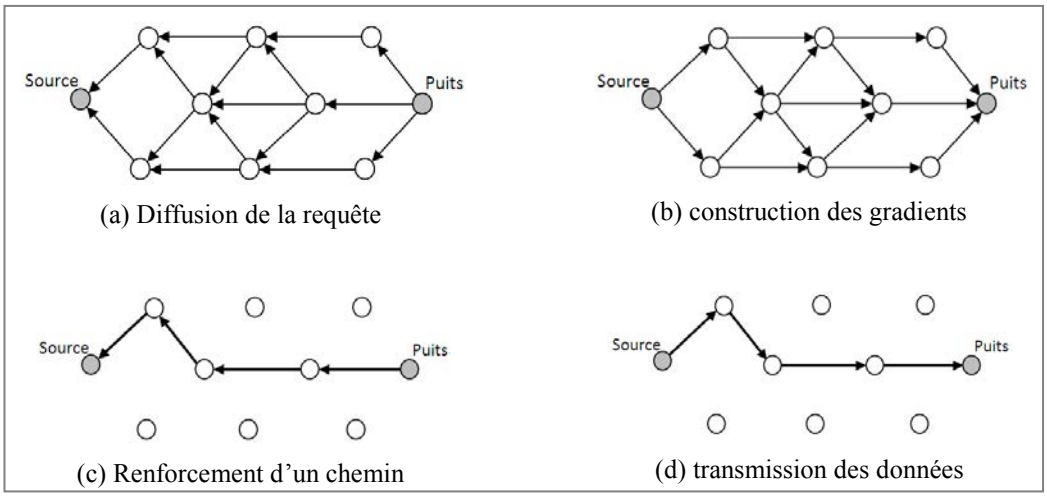


Figure 24. Directed Diffusion

DD a été amélioré par DCHT (Delay-Constrained High-Throughput Protocol for Multipath Transmission) (Li et al., 2010). DCHT adopte l'idée d'utiliser l'approche de routage multi-chemins dans les réseaux de capteurs sans fil. Il introduit une méthode de renforcement de chemin basé sur l'utilisation d'une fonction de coût de routage, qui considère le coût de transmission attendu *ETX* (Expected Transmission Count) (Couto et al., 2005) et le délai, afin de découvrir les chemins de haute qualité avec une faible latence de bout en bout entre la source et le puits.

b) EAR : L'objectif d'EAR (Energy Aware Routing) (Shah and Rabaey, 2002) est l'augmentation de la durée de vie du réseau de capteurs. Bien que ce protocole soit similaire à DD, il diffère dans le sens où il maintient une série de chemins au lieu de maintenir ou d'appliquer un seul chemin optimal renforcé à des taux de transmission élevés. Ces chemins sont entretenus et choisis en utilisant une probabilité de sélection. La valeur de cette probabilité dépend de la consommation d'énergie de chaque chemin. En utilisant différents chemins à chaque transmission de données, l'énergie de chaque chemin ne sera pas épuisée rapidement. La durée de vie du réseau peut être ainsi prolongée puisque l'énergie est dissipée d'une façon égale entre tous les nœuds.

c) GBR : GBR (Gradient Based Routing) (Schurgers and Srivastava, 2001) est similaire à DD. Son idée principale est de maintenir le nombre de sauts lorsqu'une requête est diffusée à travers le réseau. Cela implique que chaque nœud puisse découvrir le nombre minimal de sauts vers le puits appelé « hauteur du nœud ». La différence entre la hauteur d'un nœud et celle de l'un de ses voisins est appelée *gradient de lien* entre les deux nœuds.

Un paquet de réponse est transmis sur le lien avec le plus grand gradient (le plus court vers le puits). GBR utilise certaines techniques auxiliaires telles que l'agrégation des données. Lorsque plusieurs chemins de réponse passent par un nœud intermédiaire commun, ce dernier peut combiner des données par des fonctions d'agrégation.

d) BMR : BMR (Braided Multipath Routing) (Ganesan et al., 2001) a été proposé pour assurer une tolérance aux pannes pour la recherche d'information dans les réseaux de capteurs. Ce protocole utilise deux types de messages de renforcement. La construction du chemin est initiée par le puits en envoyant un message *PPR* (Primary Path Reinforcement) vers son meilleur nœud voisin en terme de qualité de service. Par exemple, sur la figure 24, le nœud puits envoie le message *PPR* au nœud *D*. De même, le nœud intermédiaire transmet le *PPR* reçu à son meilleur voisin de saut suivant vers le nœud source. Ce processus est répété jusqu'à ce que le message *PPR* atteigne la source. Tous les nœuds sur le chemin renforcé primaire construisent un chemin alternatif autour de leurs nœuds voisins de prochain saut. Chaque nœud génère un message *APR* (Alternative Path Reinforcement) et l'envoie à son deuxième meilleur nœud voisin. Par exemple, sur la figure 25, le nœud puits envoie un message *APR* au nœud G, afin d'établir un chemin de sauvegarde autour du nœud *D*. En conséquence, chaque nœud sur le trajet renforcé principal construit un chemin de secours autour de son prochain saut. Lorsque le chemin primaire ne parvient pas à transmettre des paquets de données vers le puits, les voies alternatives construites peuvent être utilisées.

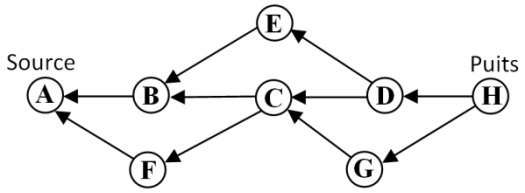


Figure 25. Phase de renforcement du protocole BMR

IV.2. Protocoles à une seule phase requête/réponse

Dans les réseaux de capteurs, la minimisation du nombre d'inondation réduit le taux de consommation d'énergie. La recherche par une seule phase requête/réponse permet d'éviter deux phases d'inondations qui existent dans les deux phases requête/réponse (données exploratoires et renforcements). Le puits diffuse par inondation une requête dans tout le réseau en établissant des gradients à chaque saut. Contrairement à un protocole à deux phases, lorsqu'une requête arrive à un nœud source, celui-ci envoie les données recherchées en remontant les chemins construits par la requête. Les chemins avec les meilleures qualités de service sont sélectionnés en priorité.

Les protocoles à une seule phase requête/réponse sont classés en : protocoles basés sur le coût de liaison entre les nœuds de capteurs, protocoles multi-chemins et protocoles basés sur l'agent de recherche. Dans la suite les différentes techniques sont présentées pour chaque classe.

IV.2.1. Protocoles basés sur le coût de liaison

a) MCFA : Le protocole MCFA (Minimum Cost Forwarding Algorithm) (Ye et al., 2001) cherche un chemin à coût minimal entre la source et le puits tout en considérant les limites des réseaux de capteurs. Chaque nœud maintient une mesure qui détermine le coût minimal vers le puits. Plusieurs mesures de coût peuvent être utilisées tel que le nombre de sauts. Pour déterminer les valeurs locales des coûts, la solution la plus simple est d'utiliser l'inondation. Initialement, le puits diffuse une requête nommée *ADV* (ADVertise) contenant un coût nul. Tous les autres nœuds initialisent leur coût à une valeur infinie. Lorsqu'un nœud reçoit un *ADV*, il vérifie si la valeur reçue additionnée au coût du lien est plus petite que la valeur locale. Dans ce cas, le nœud met à jour sa valeur locale et rediffuse *ADV*. Le relais dans MCFA n'utilise aucune identification des nœuds et aucune table de voisinage.

Le message de réponse émis par une source contient le coût minimal vers le puits. À la réception du message de réponse, le nœud vérifie si son coût local est égal au coût reçu moins le coût de lien de réception. Dans ce cas, le nœud transmet le paquet en remplaçant la valeur du coût par sa valeur locale. Ce processus continue jusqu'à atteindre le puits.

b) GRAB : Dans GRAB (GRAdient Broadcast) (Ye et al., 2005), les auteurs font l'hypothèse qu'un nœud peut estimer le coût d'envoi d'une donnée à ses voisins par l'observation du rapport signal sur bruit de transmission. Les nœuds incluent leur coût dans la requête. Le

champ sur le coût désigne le coût global vers le puits. Quand un nœud source transmet le message de réponse, il ne sélectionne pas un voisin comme prochain saut, mais il diffuse le paquet dans tout son voisinage. Seuls les voisins qui ont un coût inférieur au coût du transmetteur continuent l'acheminement du paquet de la même manière. Les paquets sont ainsi transmis de la source vers la zone de localisation du puits.

c) MARP : MARP (Sen and Ukil, 2009) est un protocole conçu pour servir des applications de quatre classes différentes de qualité de service. Quand le nœud puits nécessite des informations d'un nœud source, il inonde le réseau par une requête nommée *DATA-REQ (DATA REQuest)*. L'entête de cette requête contient un champ *ToS* (Type of Service) sur deux bits pour indiquer le type de service à assurer par la source. Par exemple, si le champ *ToS* est mis à *00*, tous les nœuds qui reçoivent un tel paquet comprennent que la requête appartient à une application normale. Selon le type d'application, la source répond au puits par un message de réponse nommée *DATA-REP (DATA REsPonse)*.

Pour une application normale qui n'a pas d'exigence de fiabilité et de contrainte de délai, la source répond au puits par un message *DATA-REP* de la manière suivante : le nœud source choisit l'ensemble des nœuds voisins qui ont le nombre de sauts le plus petit vers le puits (qui sont donc les plus proches du puits) et choisit parmi eux le nœud qui a le maximum d'énergie résiduelle. Ce travail est répété au nœud de saut suivant jusqu'à atteindre le puits. Pour une application fiable, il y a une obligation de réussite pour transmettre les paquets entre les sources et le puits mais il n'y a pas de contrainte de délai de bout en bout. Pour supporter la fiabilité, ce type d'application utilise le routage multi-chemins. Après la réception de *DATA-REQ*, la source commence la construction d'un ensemble de chemins vers le puits. Pour construire un chemin, les nœuds choisis pour le saut suivant sont les plus proches du puits en nombre de sauts et ont un haut niveau d'énergie résiduelle. Dans le cas d'une application à faible latence c'est-à-dire ayant une contrainte stricte liée au délai de transfert des paquets (application à temps critique), la source cherche les voisins qui sont les plus proches du puits en nombre de sauts, puis elle sélectionne parmi eux celui qui offre le moins de temps de latence (le nœud voisin qui a le minimum de temps d'attente pour router le paquet). La longueur de la file d'attente donne une idée sur le temps d'attente prévisible d'un paquet au niveau d'un nœud. Donc pour minimiser le temps de latence, MARP choisit les nœuds voisins qui ont les files d'attente les moins chargées.

IV.2.2. Protocoles multi-chemins

a) REAMR : REAMR (Reliable and Energy-Aware Multipath Routing) (Hassanein and Luo, 2006) est conçu pour assurer une transmission fiable des données entre la source et le puits. Le puits inonde le réseau par une requête nommée *service-path-request* qui permet de trouver tous les chemins possibles vers la source ainsi que leurs qualités de service associées. Lors de

la réception de la requête, le nœud source transmet un message *service-path-reservation* vers le puits à travers le meilleur chemin calculé par *service-path-request*. Les nœuds intermédiaires réservent une partie de leur énergie restante pour les communications qui vont avoir lieu sur ce chemin. Ensuite, le nœud source commence à transmettre ses données vers le puits à travers le chemin construit. Puis, le puits commence la construction d'un autre chemin alternatif vers la même source. Le puits émet par inondation un message *backup-path-discovery* dans le réseau, et seuls les nœuds qui ne font pas partie du chemin principal réservé participent à l'inondation de ce message. Par conséquence, un autre chemin alternatif disjoint est créé pour améliorer la tolérance aux fautes ou plus précisément à la défaillance du premier chemin. Deux chemins sont dits disjoints s'ils n'ont pas des nœuds en commun sauf la source et la destination.

b) EQSR : EQSR (Energy efficient and QoS based multi-path Routing) (Othman and Yahya, 2010) est un protocole de recherche multi-chemins avec qualité de service. La recherche d'information se fait en trois phases : phase d'initialisation, phase de construction du meilleur chemin et phase de construction du chemin alternatif. En phase d'initialisation, chaque nœud envoie un message *Hello* de découverte de voisinage. À la fin de cette phase, chaque nœud connait ses voisins avec leurs informations nécessaires (niveau d'énergie résiduelle, espace libre de file d'attente, etc.). En phase de construction du meilleur chemin, le puits commence la découverte des chemins pour trouver un ensemble de nœuds capables de transmettre les paquets à partir de la source vers le puits. Le puits choisit le meilleur voisin et lui envoie une requête pour établir un chemin vers la source. De même, le prochain nœud calcule localement son meilleur voisin et lui envoie la requête. Ce processus se déroule jusqu'à atteindre la source. À la fin de la construction du meilleur chemin, le puits construit d'autres chemins alternatifs disjoints en sélectionnant d'autres meilleurs voisins. À la fin de la construction des chemins, le protocole classe ces chemins en fonction de leurs délais de bout en bout. Les meilleurs chemins sont réservés pour le trafic à temps réel et les autres pour le trafic moins contraint.

c) MR2 : MR2 (Maximally Radio-Disjoint Multipath Routing) (Maimour, 2008) est un protocole de recherche multi-chemins. Le puits initialise le processus de recherche. Il ajoute son adresse dans la requête comme identifiant de chemin et il inonde le réseau avec ce message. Chaque fois qu'un nœud reçoit la requête, il vérifie tout d'abord s'il n'a pas l'identifiant de chemin dans sa mémoire. Dans ce cas, il ajoute le chemin à sa table de routage (il enregistre l'adresse du nœud qui a transmis la requête). Dans le cas contraire, si l'identifiant du trajet inclus dans le message reçu existe déjà dans sa table de routage, le chemin doit remplacer le chemin précédent s'il comporte un nombre de sauts inférieur. Dans ce dernier cas, le nœud doit retransmettre la requête même s'il l'a déjà fait auparavant. Ce processus se poursuit jusqu'à ce que le message de requête soit reçu par un nœud de capteurs

qui peut fournir les données demandées, auquel cas le nœud source commence la transmission des paquets vers le puits via le chemin découvert.

Afin de résoudre le problème d'interférence entre les chemins adjacents, tous les nœuds intermédiaires le long du chemin entre la source et le puits, notifient leurs voisins afin de les empêcher de participer à tout autre processus de découverte de chemin. Grâce à ce mécanisme, chaque fois qu'un chemin alternatif est construit, les nœuds autour du chemin initial ne participent pas et restent inactifs.

d) N-to-1 : L'objectif principal du protocole N-to-1 (N-to-1 Multipath Routing Protocol) (Lou, 2005) est de découvrir simultanément plusieurs chemins entre la source et le puits. Le puits commence la première étape du processus de découverte de routes par un message requête. Ce message requête utilise la technique d'inondation pour construire un arbre hiérarchique (Spanning Tree) et pour découvrir plusieurs chemins entre les nœuds et le puits. Pendant cette phase, chaque nœud qui reçoit une requête pour la première fois, enregistre l'expéditeur de ce message comme son lien vers le puits. Ce processus se poursuit jusqu'à ce que tous les nœuds de capteurs découvrent leur chemin primaire vers le puits et par conséquent un arbre hiérarchique est construit comme on peut le voir sur la figure 26(a).

La deuxième étape de ce protocole permet de découvrir plusieurs chemins de chaque nœud de capteur vers le puits. Cette phase consiste à échanger des informations sur les chemins découverts lors de la première étape entre les nœuds appartenant à différentes branches de l'arbre. Comme le montre la figure 26(b), chaque liaison entre deux nœuds appartenant à différentes branches de l'arbre peut aider à établir un chemin alternatif pour ces nœuds vers le puits. Enfin, les nœuds sources découpent leur trafic en plusieurs segments et distribuent ces segments de données sur les chemins découverts.

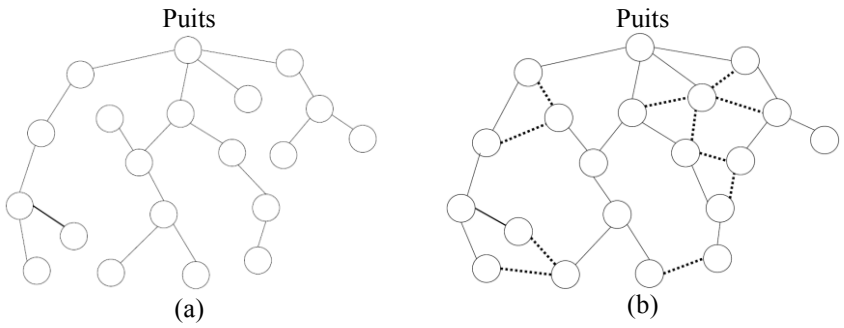


Figure 26. Construction des chemins entre sources et puits dans N-to-1

IV.2.3. Protocoles basés sur un agent de recherche

a) Rumor Routing : Comme présenté précédemment, pour chercher un ensemble d'informations dans un réseau de capteurs, l'idée générale consiste à inonder le réseau par une requête. Dans certains cas, il n'y a qu'une faible quantité de données demandées à partir de quelques nœuds ce qui rend l'utilisation de l'inondation inutile et provoque un gaspillage d'énergie. Rumor Routing (RR) (Braginsky and Estrin, 2002) résout ce problème par

l'acheminement des requêtes vers les nœuds qui ont observé un événement particulier plutôt que d'inonder tout le réseau sur chaque événement. Pour cela, RR emploie des messages appelés *agents*. Lorsqu'un nœud détecte un événement, il ajoute cet événement à sa table locale et génère un *agent*. Les agents sont acheminés dans tout le réseau dans le but de propager les informations sur les événements aux nœuds distants. Quand le puits génère une requête de recherche pour un événement, les nœuds qui connaissent la route peuvent répondre à la requête en se référant à sa table d'événements.

RR a été amélioré par DRR (Directional Rumor Routing) (Shokrzadeh et al., 2007). Il est basé sur l'hypothèse que la probabilité de croisement de deux lignes droites dans le réseau est très supérieure à celle des deux lignes cursives. Quand un nœud détecte un événement, il propage un agent à travers le réseau le long de certains chemins linéaires par localisation géographique, formant ainsi des trajectoires en forme d'étoile. Le chemin de requête est généré par le puits de la même manière dans le but de croiser un chemin d'agent pour acheminer ensuite la requête vers la source de l'événement demandé.

b) EEQR : EEQR (Energy Efficient Query-based Routing protocol) (Ahvar and Mahdavi, 2011) emploie un agent pour chercher une information. Il comporte quatre phases : découverte du voisinage, propagation de l'agent, diffusion de la requête et transmission de la réponse. Au début, chaque nœud diffuse son identifiant et son niveau d'énergie vers son voisinage par un message *Hello*. Lorsqu'un nœud reçoit ce message, il ajoute une entrée à sa liste de voisins, en mémorisant l'identifiant et le niveau d'énergie du nœud émetteur. Quand un nœud a un événement, il crée un message *Agent* et le transmet dans le réseau à plusieurs sauts. À chaque nœud, le prochain saut est choisi aléatoirement parmi les voisins dont les niveaux d'énergie sont au dessus d'un certain seuil d'énergie.

Le puits transmet la requête dans le réseau avec un champ *TTL* (Time To Live) qui définit le nombre maximal de sauts autorisés. À chaque saut, le nœud diffuse une requête nommée INFREQ (INFormation REQuest) de demande d'existence d'agent ou d'événement dans le voisinage. Si le nœud reçoit une réponse d'agent d'un nœud voisin, ce dernier sera le prochain saut de la requête. Sinon, un nœud de prochain saut est choisi aléatoirement de la même manière que l'agent. Le message de réponse est renvoyé par la source vers le puits sur le chemin inverse parcouru par la requête. Le puits fait au maximum trois tentatives de recherche. Si la première requête ne parvient pas à trouver l'information demandée, la deuxième requête puis la troisième requête sont créés. Si la troisième tentative échoue, la requête est inondée dans tout le réseau.

EEQR a été amélioré par IEEQR (Improved EEQR) (Ahvar et al., 2012). Dans IEEQR, les entêtes protocolaires d'EEQR ont été optimisés afin de réduire leur taille et réduire ainsi le temps de transmission/réception et le taux de consommation d'énergie.

IV.3. Analyse et synthèse

Le tableau 2 résume les techniques utilisées dans les protocoles de recherche d'informations présentés dans cette section. Le choix d'un protocole de recherche de données est lié à des critères propres à notre application sur la matière communicante. Dans notre cas, l'information à chercher dans le réseau, enfoui dans un préfabriqué en béton, est stockée uniformément dans différents nœuds. Suite à l'inondation d'une requête de recherche dans tout le réseau, plusieurs nœuds peuvent répondre pour une même requête. Ainsi, l'application d'un algorithme à deux phases requête/réponse aboutit à plusieurs connexions source-destination et risque d'engendrer une surcharge sur le nœud source, émetteur de la requête, et globalement conduire à une congestion du réseau. Ainsi, un protocole à deux phases requête/réponse n'est pas adapté car il risque d'épuiser rapidement les batteries des nœuds du réseau. Généralement, les protocoles de recherche à deux phases sont appliqués plus dans le contexte de flux multimédia : un puits cherche un nœud source d'un flux multimédia et sélectionne le chemin le plus fiable et avec la meilleure qualité de service pour récupérer le flux en contenu dans un réseau de capteurs.

Pour cette raison, notre choix s'est orienté vers la classe de protocoles de recherche à une seule phase requête/réponse.

En fait, les algorithmes à une seule phase requête/réponse de la littérature consistent à inonder par une requête l'ensemble du réseau et à collecter les valeurs du nœud source au niveau du nœud de départ. Comme le protocole de stockage sélectionné précédemment conduit à ce que toutes les informations soient présentes uniformément dans la matière, l'utilisation d'un algorithme par inondation de requêtes conduira à la duplication de réponses, à la congestion du réseau, à la surcharge des nœuds et finalement à l'épuisement rapide des batteries des nœuds du réseau après plusieurs phases de recherche. Une autre contrainte qui rend les protocoles de la littérature non adaptés à notre contexte de travail est que le nœud puits n'est pas connu préalablement. Comme le puits n'est pas fixe, les nœuds ne connaissent pas le coût pour l'atteindre. Ceci rend l'application d'une méthode de connaissance des coûts difficile à utiliser. En effet, dans notre contexte, le coût entre la source et la destination doit être construit au cours de l'acheminement de la requête de recherche.

Notre objectif est donc de développer un mécanisme de recherche efficace qui optimise la consommation d'énergie tout en évitant la duplication des réponses. Pour cette raison, nous avons choisi une technique de transmission dans les réseaux de capteurs nommée *Random Walk* (Tian et al., 2006) (Lima and Barros, 2007) utilisée par le protocole de stockage RaWMS (voir section III.2.2.b.) mais appliquée ici pour construire une solution de lecture. *Random Walk* consiste à transmettre des messages dans le réseau d'un nœud à l'autre (one to one). En utilisant cette technique, le nœud source de la requête choisit un nœud parmi ses voisins pour lui transmettre la requête. Ce processus se répète jusqu'à atteindre un nœud qui

contient l'information recherchée. Le *Random Walk* ne nécessite pas le maintien d'une infrastructure distribuée telle qu'un arbre hiérarchique ou clusters qui demandent du temps et de l'énergie pour la construction de la structure. Dans nos travaux, nous avons associé d'autres techniques au *Random Walk* original afin d'améliorer son efficacité et ainsi augmenter le taux de succès du processus de recherche d'une information dans la matière communicante :

- La technique *Pull-Gossip* (Liu et al., 2007) est utilisée afin d'interroger les voisins à chaque saut du *Random Walk* pour savoir s'ils ont l'information recherchée.
- Le nœud de chaque prochain saut du *Random Walk* est choisi parmi les nœuds de bordure de voisinage afin de couvrir plus de nouveaux nœuds dans le processus *Pull Gossip*.
- Une fonction d'évaluation de la qualité de service a été proposée afin d'acheminer la requête sur un chemin à haute qualité de service.
- Un paramètre *TTL* (Time To Live) est utilisé pour limiter la longueur du chemin de recherche. Cela évite les boucles de recherche infinie dans le cas où le chemin ne conduit à rien.

Toutes ces techniques sont détaillées dans chapitre suivant. Le protocole de recherche développé est nommé ***RaWPG*** : *Random Walk and Pull Gossip data retrieval protocol*. Il est présenté dans le chapitre suivant et évalué par simulation au dernier chapitre de ce rapport.

Tableau 2. Résumé des protocoles de recherche d'informations dans les réseaux de capteurs sans fil

			Stratégies de recherche			Techniques de recherche				
Protocole de recherche	**Deux phases requête/réponse**	**Une seule phase requête/réponse**	**Agent de recherche**	**Multi-chemins**	**Coût de liaison**	**Inondation de la requête**	**Agrégation**	**Qualité de service**	**Optimisation d'énergie**	**Arbre hiérarchique**
Directed Diffusion	✓					✓				
DCHT	✓			✓		✓		✓		
EAR	✓			✓		✓			✓	
GBR	✓					✓	✓			
BMR	✓			✓		✓		✓		
MCFA		✓			✓	✓				
GRAB		✓			✓	✓				
MARP		✓			✓	✓		✓		
REAMR		✓		✓		✓		✓		
EQSR		✓		✓		✓		✓		
MR2		✓		✓		✓				
N-to-1		✓		✓		✓				✓
Rumor Routing		✓	✓							
DRR		✓	✓							
EEQR		✓	✓						✓	
IEEQR		✓	✓					✓	✓	

V. Conclusion

Dans ce chapitre, les différentes techniques de stockage et de recherche d'information ont été expliquées à travers plusieurs protocoles de communication de la littérature de façon à analyser leurs intérêts et leurs limites dans notre contexte applicatif. Dans la suite de ce mémoire, nous allons nous attacher à détailler nos solutions de stockage et de lecture en proposant des modèles et des protocoles. Ces solutions seront ensuite implémentées et simulées pour évaluer leurs performances globales vis-à-vis des critères comme l'uniformité de réplication, l'efficacité de stockage et la fiabilité de lecture.

CHAPITRE III

Spécification de protocoles de stockage et de lecture d'informations dans la matière communicante

I. Introduction

Après avoir examiné dans le chapitre 2, les travaux de la littérature sur les protocoles de stockage et de lecture d'informations dans les réseaux de capteurs sans fil, nous allons proposer nos propres solutions pour les adapter à la gestion et à la surveillance de préfabriqués en béton durant tout leur cycle de vie. Dans ce contexte, les nœuds de capteurs sont dispersés régulièrement dans le béton, et chaque nœud peut communiquer sans fil avec ses nœuds voisins dans son champ de transmission radio (réseau connexe). Donc, chaque nœud peut atteindre indirectement tous les autres nœuds dans le préfabriqué à travers une communication multi-sauts. Dans ce chapitre, nous présentons dans un premier temps la notion de nœud maitre. Dans une deuxième partie, nous allons concevoir le protocole de dissémination pour un stockage uniforme d'information dans la matière communicante. Finalement, nous allons spécifier le protocole de lecture d'information adapté à la solution de stockage, en détaillant tous les mécanismes adoptés, les entêtes protocolaires des messages échangés et le comportement du nœud lors de la réception de chaque type de message.

II. Notion de noeud maitre

Lorsque l'utilisateur veut communiquer avec un préfabriqué pour des opérations de stockage et de lecture, il choisit un nœud dans le béton qui se trouve dans son champ de transmission radio comme le montre la figure 27. Ce nœud sélectionné est nommé « nœud maître » dans cette thèse. N'importe quel nœud dans le préfabriqué peut être un « maître », il dépend du choix de l'utilisateur lors de la connexion.

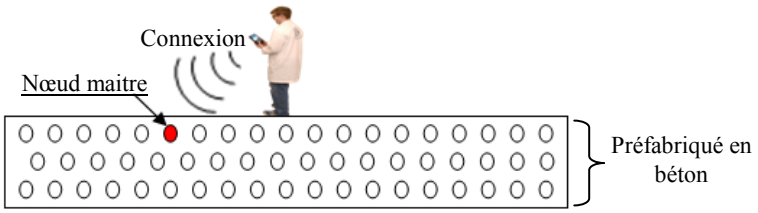


Figure 27. Connexion de l'utilisateur à la matière par le nœud maitre

La figure 28 présente le diagramme de séquence d'interaction utilisateur/matière pour la sélection du nœud maitre, le stockage et la lecture d'information à travers ce nœud. L'utilisateur utilise un dispositif approprié pour se connecter à la matière béton. Lorsque l'utilisateur commence la connexion, le dispositif diffuse un message de découverte dans son voisinage. Tous les nœuds dans le champ de transmission radio de l'utilisateur répondent chacun par un message qui contient leurs caractéristiques matérielles comme l'énergie restante, la taille de la file d'attente, etc. L'appareil choisit automatiquement le nœud le plus adapté (comme par exemple le nœud avec l'énergie résiduelle la plus élevée). L'utilisateur peut ensuite commencer le processus de stockage ou de lecture. Comme illustré sur la figure

28, la phase de lecture n'est pas nécessairement précédée par une phase de stockage. Cependant, ces deux phases sont toujours précédées d'une identification d'un nœud maitre.

Dans ce qui suit, les processus de stockage et de lecture d'informations dans une matière communicante sont détaillés.

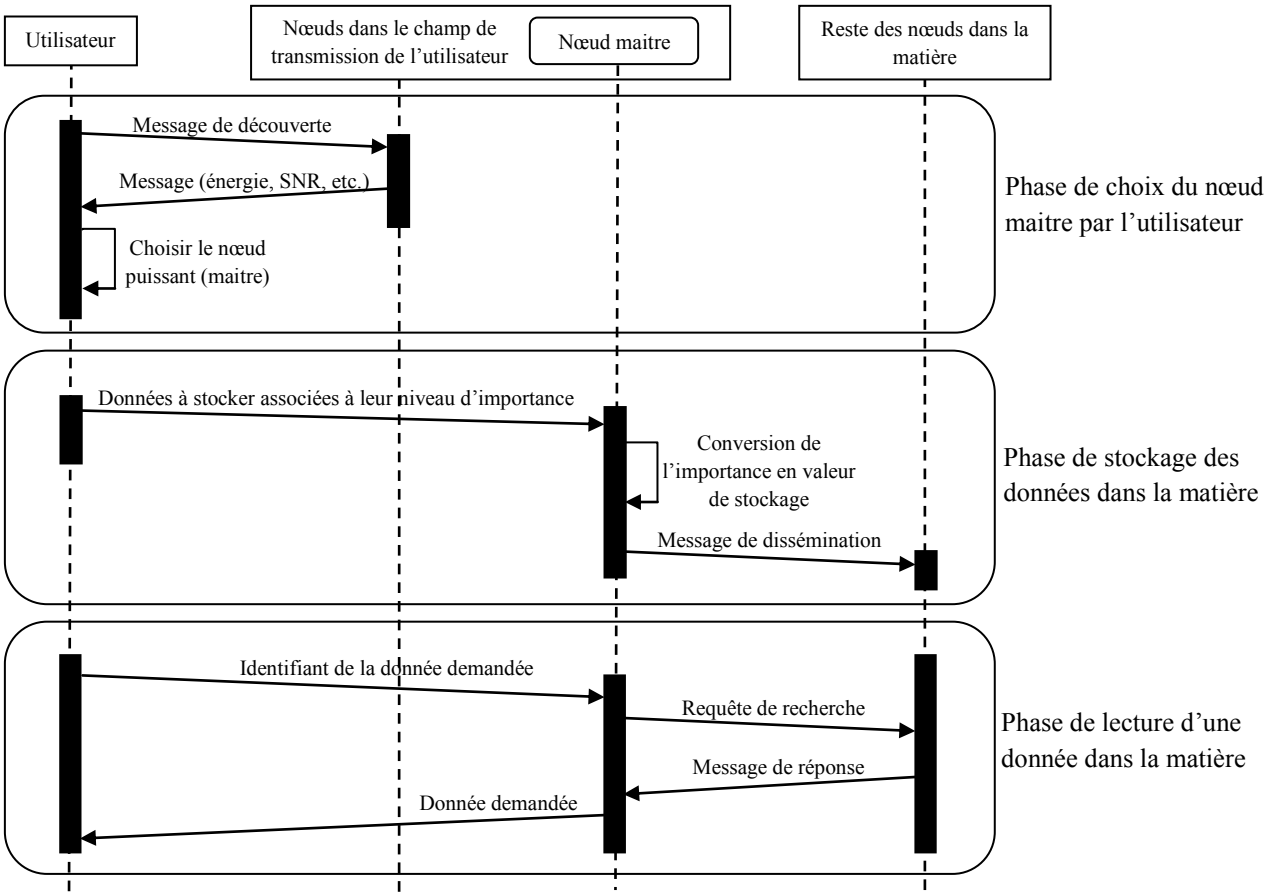


Figure 28. Diagramme de séquence d'interaction utilisateur/matériel pour le stockage et la lecture

III. Stockage des informations dans la matière communicante

Ce processus est réalisé grâce au développement d'un protocole de communication entre les nœuds de capteurs enfouis dans la matière (le béton pour notre application). Le protocole est nommé USEE (Uniform Storage and Energy Efficient data dissemination protocol) présenté dans (Mekki et al., 2016b). USEE est un protocole de dissémination proactif non structuré basé sur un mécanisme d'inondation (afin de transmettre la donnée vers tous les nœuds de la matière communicante) et sur une technique de stockage pour répliquer la donnée dans différents nœuds de la matière (voir la figure 29). Dans la suite de ce chapitre, nous présentons les mécanismes d'inondation et les mécanismes de stockage existants dans la littérature pour justifier nos choix dans le développement de notre protocole USEE.

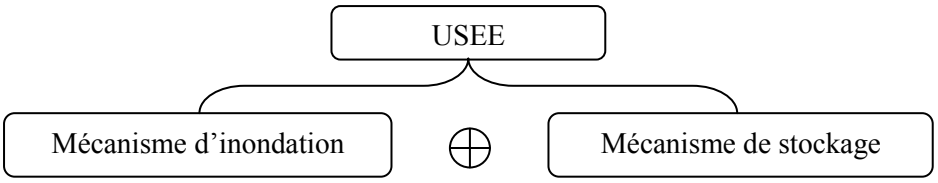


Figure 29. Structuration du protocole de dissémination USEE

III.1. Inondation dans les réseaux de capteurs sans fil

L'inondation (transmission d'un message d'un nœud source à tous les nœuds du réseau) a un rôle important dans les réseaux de capteurs sans fil. Cette technique est généralement utilisée pour la découverte des chemins pour acheminer des données et pour la reprogrammation des nœuds du réseau.

L'inondation (en anglais : *Flooding*) est un algorithme dans lequel chaque nœud du réseau retransmet un message, reçu pour la première fois, dans son voisinage. Ce mécanisme permet la transmission multi-sauts d'un message vers tous les nœuds dans le réseau.

III.1.1. Techniques d'inondation

Le *Flooding* produit une quantité excessive de trafic redondant (réception d'un même message plusieurs fois par chaque nœud) en raison de transmissions de messages par tous les nœuds dans la même période de temps. Ce comportement provoque une congestion du réseau et une grande consommation d'énergie. Ainsi, si les voisins d'un nœud sont nombreux, un grand taux d'interférences et de collisions aura lieu (Sanchez et al., 2011). Pour résoudre ces problèmes, différentes améliorations de *Flooding* ont été proposées comme indiqué dans le tableau 3. Ces techniques sont généralement divisées en deux catégories: *déterministe* et *non déterministe*.

Les techniques déterministes nécessitent des informations sur la topologie globale/locale du réseau pour la prise de décision lors de la rediffusion du message. Par opposition, les techniques non déterministes ne requièrent pas d'informations topologiques. Chaque nœud étant autorisé à retransmettre un message selon une probabilité de transmission prédéterminée.

Tableau 3. Techniques d'inondation dans les réseaux de capteurs sans fil

	Mécanismes d'inondation
Déterministe	Distance-based (Yubo et al., 2009) Location-based (Chang et al., 2005) Cluster-based (Khiati and Djenouri, 2012)
Non déterministe	Probabilistic-based (Busnel et al., 2007) Counter-based (Aminu et al., 2007)

a) Distance-based : Ce mécanisme est basé sur le fait que si la distance entre deux nœuds *A* et *B* est très faible, il y a peu de couverture supplémentaire si le nœud *B* rediffuse le message. Par contre si la distance est plus grande, la couverture supplémentaire sera plus large. Donc,

lorsqu'un message est reçu pour la première fois par le nœud *B*, la distance ||AB|| détermine la décision de diffusion. Si ||AB|| est inférieure à un seuil *d*, le message d'inondation est abandonné par *B*. Sinon, le message est transmis après la fin d'un délai d'attente aléatoire à la condition que le même message n'ait pas été reçu de la part d'un nœud *C* où ||BC||<*d*. Pour calculer les distances entre les nœuds, ces méthodes utilisent des techniques d'estimation de distance (comme par exemple ToA et RSS).

b) Location-based : Ce mécanisme emploie les informations de position géographique en utilisant par exemple le GPS pour évaluer la couverture supplémentaire lors d'une rediffusion d'un message. Lors de la réception d'un message, le nœud calcule la couverture supplémentaire en se basant sur les positions géographiques des nœuds. La couverture supplémentaire représente une valeur dans l'intervalle [0, 1] suite à une division par πr^2 (la couverture maximale) où *r* est le rayon du champ de transmission radio du nœud. Si la couverture est inférieure à un seuil fixé dans l'intervalle [0, 0.61], le message est abandonné, sinon le message est mis en mémoire et un délai d'attente aléatoire est démarré (Chang et al., 2005). Le message est ensuite transmis après l'expiration du délai.

c) Cluster-based : Ce mécanisme nécessite un algorithme de *clustering*, c'est-à-dire de partitionnement du réseau, où tous les chefs de cluster peuvent couvrir tous les nœuds dans leur zone. Seuls les chefs de cluster diffusent le message dans leur zone vers leurs nœuds membres. Ce mécanisme réduit la probabilité de collision et la redondance puisqu'un seul nœud est responsable de la retransmission dans chaque zone. De plus, pour propager le message aux nœuds membres dans d'autres clusters, des nœuds passerelles sont employés.

d) Probabilistic-based : Le mécanisme probabiliste définit une valeur *p*. À la réception d'un message, le nœud rediffuse le message avec la probabilité *p*. Lorsque *p*=1, ce mécanisme est équivalent à un simple *Flooding*. Cette technique réduit la redondance en utilisant des petites valeurs de *p*. Un temps d'attente aléatoire est inséré avant la rediffusion du message pour différencier l'instant de transmission entre les nœuds voisins et réduire ainsi le taux de collision dans le réseau.

e) Counter-based : La technique counter-based emploie une méthode différente pour réduire la redondance et l'accès concurrent. Lorsque le nœud reçoit un message, il fixe un délai d'attente aléatoire. Pendant ce délai, le nœud compte le nombre de retransmissions du même message par ses nœuds voisins. Après l'expiration du délai, le message est diffusé seulement si le nombre de retransmissions calculé est inférieur à un seuil prédéterminé.

f) Analyse et critères de choix

Il est clair que les techniques basées sur la distance et la localisation exploitent respectivement la distance entre les nœuds et leurs positions géographiques. Ces informations sont utiles pour réduire la quantité de retransmissions dans chaque voisinage. Cependant, les nœuds doivent être équipés d'un indicateur de force du signal reçu (RSSI) ou d'un système de

positionnement global (GPS). Ces équipements matériels supplémentaires augmentent le coût et la taille des nœuds.

Avec la diffusion basée sur le *clustering*, le réseau est divisé en groupes de nœuds. Cette technique nécessite une phase de formation des clusters et de maintenance continue des clusters à l'aide d'algorithmes spécifiques qui consomment de l'énergie supplémentaire.

Les techniques probabilistic-based et counter-based réduisent les retransmissions inutiles d'un simple *Flooding*. Ce résultat est obtenu sans matériel supplémentaire et sans aucune communication relative à la maintenance des clusters. Le coût des nœuds peut être réduit et les ressources d'énergie peuvent être ainsi conservées. Ces caractéristiques sont favorables pour les réseaux de micro-nœuds de capteurs (Tran-Dang et al., 2014). Cependant, les auteurs dans (Garbinato et al., 2010) (Izumi et al., 2007) ont montré que la technique counter-based surpasse celles étant probabilistic-based en termes d'accessibilité (augmentation du nombre des nœuds ayant reçu le message à la fin du processus d'inondation) et d'efficacité énergétique (réduction du nombre total de transmissions pour inonder un message dans tout le réseau). Pour cette raison, le counter-based peut être considéré comme un algorithme de diffusion prometteur pour les réseaux de micro-nœuds de capteurs sans fil pour nos applications sur la matière communicante.

III.1.2. Mécanisme de diffusion « counter-based »

Suite à l'analyse effectuée, nous avons doté USEE d'un mécanisme d'inondation counter-based pour transmettre efficacement les messages dans la matière. L'algorithme de diffusion counter-based est détaillé dans ce qui suit :

1. Lorsqu'un nœud reçoit un message pour la première fois, le nœud initialise un compteur N à 1, et choisit un temps d'attente aléatoire *RAD* (Random Assessment Delay) parmi les valeurs comprises entre 0 et T_{max}.
2. Si le nœud reçoit le même message pendant *RAD*, il incrémente le compteur N ($N \leftarrow N+1$). Si le compteur atteint un seuil prédéterminé *Cth*, il annule la rediffusion de message.
3. Après l'expiration du temps *RAD*, le nœud diffuse le message dans son voisinage (retransmet le message à tous ses nœuds voisins).

Discussion sur *Cth* et T_{max} : Accroître *Cth* donne une meilleure accessibilité, mais réduit également l'efficacité (retransmission par un grand nombre de nœuds) (Jacobsson et al., 2011). Les résultats théoriques dans (Machado, 2005) suggèrent que la valeur de *Cth* devrait être inférieure ou égale à 4 pour une bonne performance dans les réseaux de capteurs. Cependant, les auteurs dans (Arango et al., 2006) montrent qu'un seuil *Cth* entre 4 et 6 permet un bon compromis entre accessibilité et efficacité. En outre, les résultats de simulation de (Izumi et al., 2008) (Izumi et al., 2007) montrent que si le seuil est fixé à 4, l'accessibilité est supérieure à

99,5%. Dans (Jacobsson et al., 2011), le mécanisme a été testé dans un réseau de capteurs réel en utilisant les nœuds *t-Mote-Sky* de Moteiv-Corporation. Les auteurs ont montré qu'un seuil de 2 provoque un faible taux d'accessibilité, alors qu'une valeur supérieure à 3 ne l'améliore pas beaucoup et ne fait qu'augmenter le nombre de retransmissions et le taux de consommation d'énergie moyenne. Ils ont montré aussi que lorsque T_{max} est trop petit, les retransmissions augmentent. Par ailleurs, lorsque le *RAD* dépasse les 200 ms, il y a très peu d'amélioration en efficacité. Les auteurs ont montré aussi que *Cth*=3 et T_{max}=200ms donnent de bons résultats en terme d'accessibilité.

III.2. Stratégies de stockage dans les réseaux de capteurs sans fil

Le tableau 4 donne deux techniques de stockage qui sont employées dans les protocoles de dissémination proactif non structurés pour les réseaux de capteurs : le *hop-counter-based* et le *probabilitic-based*.

Tableau 4. Techniques de stockage dans les réseaux de capteurs sans fil

	Mécanismes de stockage
Déterministe	Hop-counter (Bar-Yossef et al., 2008) (Mekki et al., 2013)
Non déterministe	Probabilistic-based (Vecchio et al., 2010) (Viana et al., 2010) (Maria et al., 2013)

Le stockage *hop-counter* consiste à utiliser un compteur du nombre de sauts présenté sous forme d'un champ dans l'entête d'un message. Il est fixé au début à un nombre entier précisant le nombre de sauts à effectuer avant de stocker l'information. Le message est transmis d'un nœud à un autre, et à chaque saut, le compteur est décrémenté. Les nœuds qui reçoivent un compteur de valeur zéro, stockent l'information portée par le message dans leur mémoire interne. Puis, le nœud remet le compteur à sa valeur initiale et retransmet le message vers ses voisins qui n'ont pas encore reçu ce message pour continuer le processus de dissémination.

Le stockage probabiliste adopte une autre stratégie. Lors de la réception d'un message transmis dans le réseau, chaque nœud stocke l'information avec une probabilité *P*. Par exemple, si une probabilité de stockage est égale à 0.5 et que le nœud choisit un nombre aléatoire entre 0 et 1 alors si ce nombre est inférieur ou égal à 0.5, le nœud stocke l'information dans sa mémoire interne. Sinon, l'information n'est pas stockée.

Une étude comparative entre ces deux techniques de stockage a été faite dans (Mekki et al., 2016a) (voir Annexe). Les résultats ont montré que le stockage probabiliste permet une réplication plus uniforme de l'information dans la matière alors qu'avec la technique *hop-counter*, des zones dans la matière sont vides d'informations. Le stockage probabiliste a donc été adopté pour notre solution.

III.3. Le protocole de dissémination USEE

Dans ce qui suit, nous présentons le protocole de dissémination USEE en détaillant le modèle de fonctionnement des nœuds, l'entête protocolaire du message échangé et l'algorithme.

III.3.1. Etats des nœuds de capteurs dans le processus de dissémination

Chaque nœud a quatre états au cours de la dissémination d'une donnée dans le réseau de capteurs: ① Mode inactif, ② Mode de réception, ③ Mode de stockage et ④ Mode de diffusion.

Le passage d'un état à un autre est donné par la figure 30. Le nœud maître est présenté par la couleur rouge. Tout d'abord, tous les nœuds sont dans l'état inactif (état 1 de la figure 30(a)). Les nœuds sont non actifs et leurs modules radio sont en mode veille pour économiser leur batterie. Le nœud maître commence à diffuser le message dans son voisinage. De fait, tous les nœuds voisins passent alors dans le mode réception (état 2 de la figure 30(b)). En mode réception, le nœud reçoit la donnée et son niveau d'importance associé. Ensuite, ces nœuds passent en mode stockage (état 3 de la figure 30(c)). Sur la base du résultat de la fonction de décision de stockage, le nœud détermine s'il stocke les informations ou non. Quel que soit le résultat de la décision, le nœud entre dans le mode de diffusion, pendant lequel il transmet le message à ses voisins (état 4 de la figure 30(d)). Dans une dernière étape, le nœud retourne à l'état inactif (état 1 dans la figure 30(e)). Les nœuds voisins suivants continuent le même processus (état 2 de la figure 30(d), état 3 de la figure 30(e) et état 4 de la figure 30(f)) jusqu'à ce que tous les nœuds dans la matière reçoivent et traitent le message. Le passage d'un mode à un autre est détaillé par un modèle Réseau de Petri dans la section suivante.

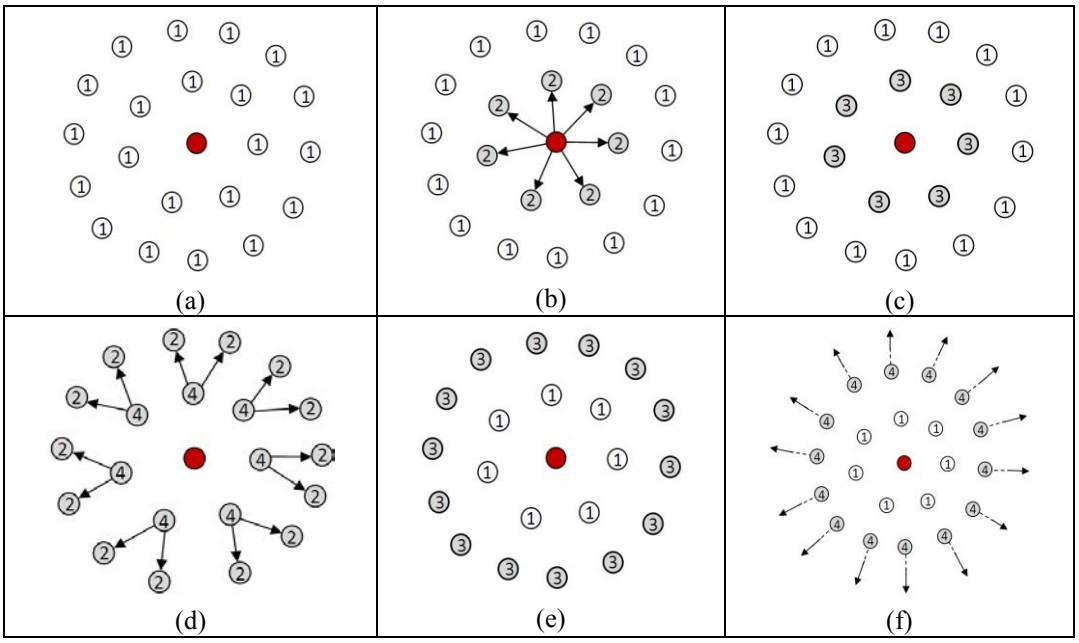


Figure 30. Variation des états des nœuds pour la dissémination des informations par USEE

III.3.2. Modèlisation par Réseau de Petri d'un noeud dans le processus de dissémination

La modélisation par Réseau de Petri (RdP) permet de créer des modèles hiérarchiques afin de simplifier l'interaction complexe et de le diviser en d'autres sous-modèles (Zouinkhi et al., 2011). Le modèle hiérarchique d'un Réseau de Petri contient des états et des transitions représentant un autre sous-modèle. La figure 31 présente le modèle de dissémination de chaque nœud dans la matière. Comme illustré, le changement d'un état à un autre d'un nœud est lié à la transition et l'apparition d'un événement. Le modèle est séquentiel, le nœud passe d'un état à un autre que si la tâche précédente est terminée. Au début, le nœud quitte l'état inactif (état p1) vers le mode de réception (état p2). Le nœud commence le mode diffusion (état p4) après la fin du stockage (état de p3). Après la diffusion du message (transition t4), le nœud retourne en état inactif.

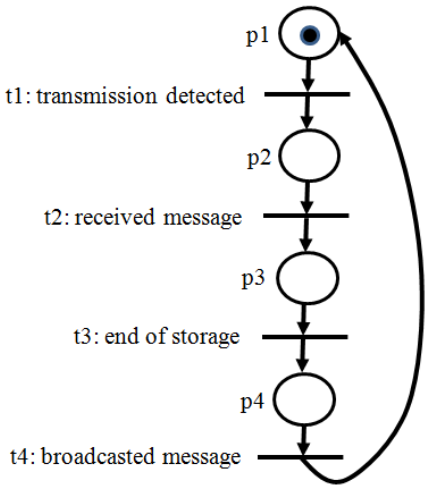


Figure 31. Modèle RdP d'un noeud dans un processus de dissémination

Le RdP de la figure 32 montre la communication entre les nœuds de la matière pour accomplir le processus de dissémination dans le réseau. L'utilisateur envoie le message au nœud maître qui commence la diffusion vers le reste du réseau par un processus multi-sauts (le message passe d'un voisin à un autre) comme décrit précédemment. Chaque fois qu'un nœud reçoit le message, il exécute tous les états de p2 à p4, puis il transmet le message aux nœuds voisins (grâce à la propagation du signal radio). Ce processus se poursuit jusqu'à ce que le message atteigne tous les nœuds du réseau dans la matière. Dans le reste de cette section, chaque mode est détaillé.

a) Mode de réception : Dans cet état, le nœud reçoit le message de l'un de ses voisins. Le message reçu contient les données et leur niveau d'importance associé. La communication entre les noeuds voisins est basée sur la diffusion. Avec un tel modèle de communication, le message peut être traité par des nœuds qui l'ont déjà reçu précédemment. Le message doit alors être identifié, ce qui permet aux nœuds de n'avoir à traiter le même message qu'une seule fois. Si un nœud reçoit à nouveau un message traité avec le même identifiant, celui-ci

est abandonné et le nœud se remet en mode inactif (état p1) comme l'illustre le modèle RdP de la figure 33.

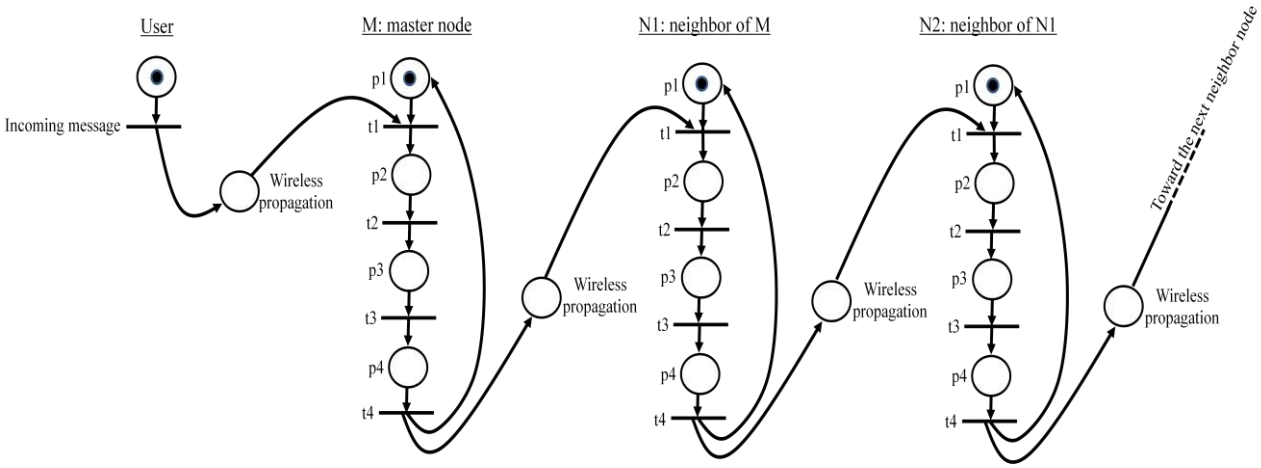


Figure 32. Modèle RdP du processus de dissémination de données dans la matière communicante

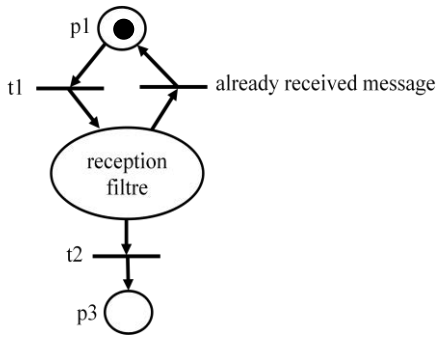


Figure 33. Modèle RdP du mode réception (p2)

b) Mode de stockage : Après réception d'un message non redondant (message reçu pour la première fois), le nœud décide de stocker ou pas l'information. La décision de stockage dépend de trois paramètres : la mémoire disponible, le niveau d'importance de la donnée et la décision de stockage pris par ses nœuds voisins. Tout d'abord, le nœud vérifie s'il possède une mémoire suffisante pour stocker la donnée reçue. Puis, il fixe un délai d'attente et se met en écoute de messages transmis dans son voisinage, et vérifie qu'aucun voisin n'a stocké la même donnée durant ce délai. Pour ce faire, un champ binaire nommé *NS* est défini dans l'en-tête du message. Il est mis à 1 si le noeud a stocké la donnée. De cette manière, le voisin peut savoir qu'il reçoit un message d'un nœud ayant stocké ou non cette donnée. Enfin, nous avons utilisé le niveau d'importance *I* (valeur compris entre 0 et 1) en tant que dernière étape pour décider le stockage. Ce niveau est considéré comme une probabilité de stockage *P=I* (stockage probabiliste) comme illustré par la fonction donnée sur la figure 34. Ainsi, le noeud stocke les données avec une probabilité *P* : il sélectionne une valeur aléatoire entre 0 et 1, si la valeur est inférieure ou égale à la probabilité *P*, la donnée est stockée.

En conclusion, le noeud stocke une donnée uniquement s'il a suffisamment de mémoire disponible, si aucun nœud voisin n'a stocké la même donnée pendant un délai d'attente, et si

la valeur aléatoire sélectionnée est inférieure au niveau d'importance (probabilité de stockage). Si le nœud stocke une donnée dans sa mémoire, il met le champ d'entête du message *NS* à 1 et il diffuse le message même si le nœud décide de ne pas diffuser le message dans le mode de diffusion. Si le nœud ne stocke pas la donnée, le champ *NS* est laissé à 0 (voir RdP de la figure 35).

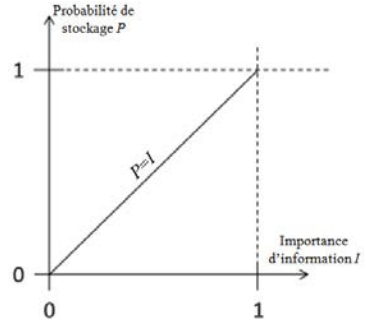


Figure 34. Conversion de l'importance d'information *I* en probabilité de stockage *P*

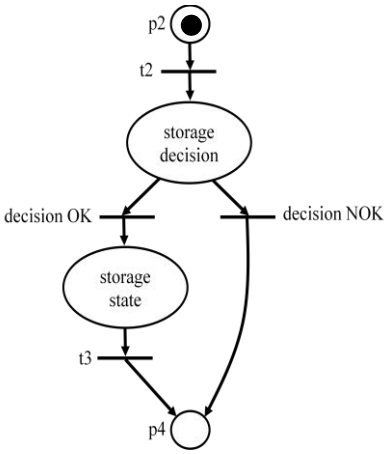


Figure 35. Modèle RdP du mode stockage (p3)

c) Mode de diffusion : Ce mode permet aux noeuds de diffuser le message dans leur voisinage pour continuer le processus de dissémination vers les nœuds qui n'ont pas encore reçu le message. La décision de diffusion utilise le mécanisme d'inondation *counter-based*. Le noeud fixe le délai d'attente *RAD* et compte le nombre d'envois du même message (duplication de message) dans son voisinage. Le message est diffusé uniquement si le nombre est inférieur à un seuil prédéterminé *Cth*. Sinon, le noeud abandonne le message et se remet en état inactif comme le montre le modèle RdP de la figure 36.

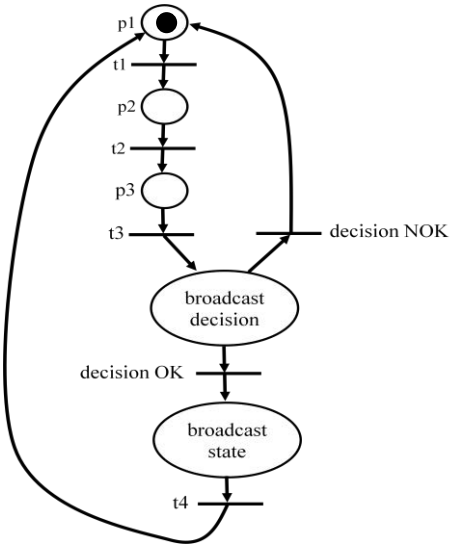


Figure 36. Modèle RdP du mode diffusion (p4)

Le modèle RdP complet de dissémination de chaque nœud dans la matière communicante est illustré par la figure 37.

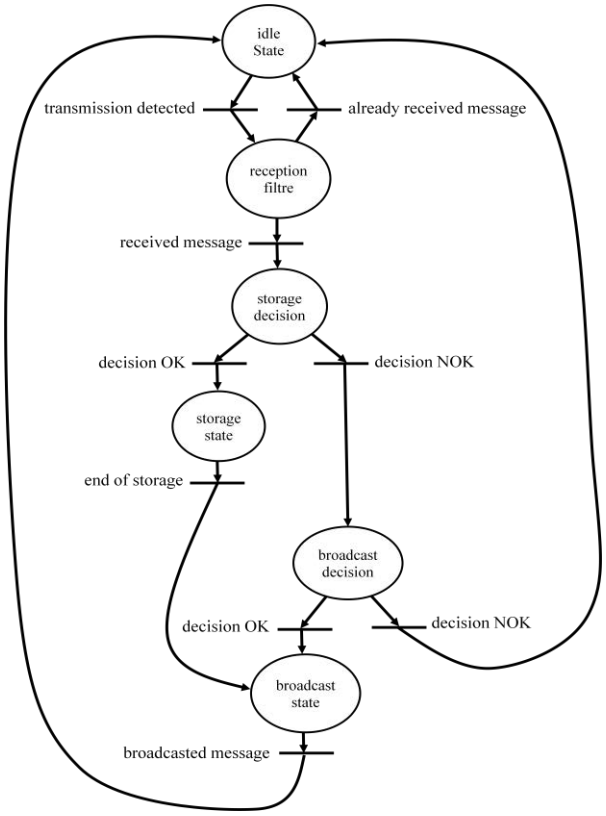


Figure 37. Modèle RdP complet d'un nœud dans le processus de dissémination USEE

III.3.3. Entête de message

La structure du message du protocole de dissémination USEE est présentée dans la figure 38. Le message nommé DATA-MSG se compose de sept champs, six sont réservés à l'entête et le dernier contient la donnée à stocker. Chaque champ est décrit dans ce qui suit :

- *Type* : Il est utilisé pour savoir quels champs composent le message.

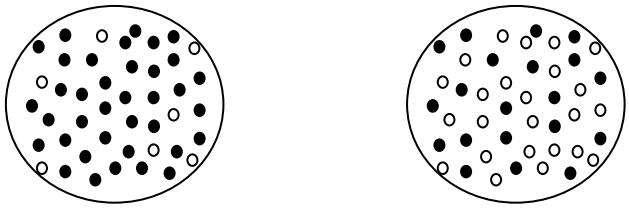
- *Size* : Il indique la taille des données contenues dans le champ "*Data*" (7ème champ).
- *Seq_Number* : Il fournit le numéro de séquence du message en cours.
- *Timestamp* : C'est l'identifiant de la phase de dissémination. Il indique la date de la dissémination de l'information par le nœud maitre. Plusieurs messages peuvent avoir le même *Timestamp*, mais le couple *Timestamp* / *Seq_Number* est unique. Ce couple est utilisé comme solution pour identifier le message et pour gérer la duplication au cours de la dissémination comme cela est décrit précédemment.
- *Probability* : Il fournit la probabilité *P* de stockage qui sera utilisée par chaque nœud pour répliquer les données dans sa mémoire interne.
- *Node_Storage (NS)* : C'est un champ d'un bit qui prend deux valeurs 0 ou 1. *NS* est utilisé pour contrôler le taux de réplication du message dans chaque voisinage.
- *Data*: Le contenu des données est inséré dans ce champ.

Type	Size	Seq_Number	Timestamp	Probability	NS	Data

Figure 38. Structure du message DATA-MSG

III.3.4. Algorithme de dissémination

Le mécanisme de diffusion *counter-based* est utilisé comme base pour intégrer le mécanisme de stockage. Pour assurer une dissémination uniforme et un taux de réplication contrôlé des données (résilience à la surcharge de la mémoire de la matière), le champ de l'entête *NS* est utilisé. Quand un nœud stocke une donnée, il retransmet le message avec *NS* = 1 pour informer ses nœuds voisins. Sinon, *NS* est mis à 0. Ce mécanisme permet d'éviter la réplication inutile dans un voisinage en gardant l'uniformité dans tout le réseau. Au lieu de stocker l'information dans plusieurs nœuds dans le voisinage (figure 39(a)), le protocole réduit ce nombre (figure 39(b)).



(a) réplication non contrôlé (b) réplication contrôlé

Figure 39. Effet du stockage contrôlé dans un voisinage

L'algorithme du protocole de dissémination USEE fonctionne comme suit (voir *Algorithme1*) : l'algorithme USEE est basé sur le mécanisme d'inondation. Avec un tel modèle de communication, le message peut être traité par les nœuds qui ont déjà reçu le même message. Le couple *Timestamp/Seq_Number* est utilisé comme identifiant du message, ensuite les nœuds sont limités à traiter le même message qu'une seule fois. Si un nœud reçoit de nouveau un message traité avec le même identifiant, il sera abandonné (lignes 1 et 2). Pour cela, lors de la première réception d'un message, le nœud vérifie la valeur *NS*. Si *NS* = 1 (le nœud reçoit le

message à partir d'un voisin qui a stocké la donnée), la décision de stockage est annulée. Dans ce cas, le nœud réinitialise *NS* à 0, initialise un compteur *N* à 1 et fixe un délai d'attente *RAD* à une valeur comprise entre 0 et T_{max} et attend son expiration (lignes 3 à 7). En attendant, le nœud compte le nombre *N* du message dupliqué reçu (incrémentation de *N* à chaque message dupliqué reçu). Après l'expiration de *RAD*, le nœud utilise la procédure BROADCAST_DECISION pour décider la diffusion du message (lignes 28 à 34). Le message est diffusé uniquement si *N* est inférieur au seuil *Cth*.

Si *NS*=0 lors de la première réception d'un message (le nœud reçoit le message à partir d'un voisin qui n'a pas stocké la donnée), le nœud initialise un compteur *N* à 1, fixe un délai d'attente *RAD* à une valeur comprise entre 0 et T_{max} et attend son expiration (lignes 8 et 9). En attendant, le nœud compte le nombre *N* du message dupliqué reçu. Lorsque *RAD* expire, le nœud vérifie s'il a reçu un message dupliqué qui contient *NS* = 1 (cela signifie que pendant l'attente d'expiration de *RAD*, un nœud voisin a déjà stocké la donnée). Dans ce cas, la décision de stockage est annulée (lignes 10 et 11). Le nœud continue alors vers la décision de diffusion en utilisant la procédure BROADCAST_DECISION.

Si le nœud ne reçoit aucun message dupliqué avec *NS* = 1 avant l'expiration de *RAD*, le nœud continue vers la décision de stockage. Tout d'abord, il vérifie la taille de la mémoire disponible (lignes 12-14). Si la mémoire est pleine, le nœud annule le stockage et passe à la procédure de BROADCAST_DECISION.

Si le nœud a suffisamment de mémoire disponible, le nœud utilise la probabilité *P* (égal au niveau d'importance *I* (*P* = *I*)) pour décider de stocker la donnée ou non. Par exemple, si *P* = *I* = 0.5 et que le nœud sélectionne une valeur aléatoire *rand* dans l'intervalle [0,1] alors le nœud stocke la donnée uniquement si $rand \leq P$(= 0,5), puis il met *NS* à 1 et il diffuse le message sans utiliser BROADCAST_DECISION, même si le seuil *Cth* de mode de diffusion est atteint (lignes 15-20). Nous choisissons de diffuser directement le message après un stockage de donnée, pour informer les voisins de cette décision, autant que possible, de sorte que les voisins annulent leur stockage pour éviter la réplication inutile dans le voisinage. Si nous utilisions la procédure BROADCAST_DECISION, la décision de diffusion pourrait conduire à ne pas diffuser le message, et ainsi les voisins ne seraient pas informés de la décision de stockage. Enfin, si $rand > P$, le nœud ne stocke pas la donnée et passe à la procédure BROADCAST_DECISION pour décider de diffuser le message ou non (lignes 21 et 22). Il est clair que le nœud met *NS* à 1 si et seulement si le nœud décide de stocker la donnée, sinon *NS* est laissé à la valeur 0.

Algorithme1: USEE

For a node *X*: upon reception of a *DATA-MESSAGE* message, do:

```
 1: if DATA-MESSAGE is already received, NUM_SEQUENCE / TIMESTAMP is known
 2:     - Drop DATA-MSG
 3: else
 4:     if DATA-MSG has NS=1
 5:         - NS←0
 6:         - Set RAD and wait for it to expire. While waiting, count the number N of the
            received duplicate DATA-MSG.
 7:         - BROADCAST_DECISION(N, DATA-MSG)
 8:     else
 9:         - Set RAD and wait for it to expire. While waiting, count the number N of the
            received duplicate DATA-MSG.
10:         if duplicated DATA-MSG has NS=1
11:             - BROADCAST_DECISION(N, DATA-MSG)
12:         else
13:             if the memory is full
14:                 - BROADCAST_DECISION(N, DATA-MSG)
15:             else
16:                 - Select random value rand between 0 et 1
17:                 if rand ≤ P
18:                     - Store the data item in the memory
19:                     - NS←1
20:                     - Broadcast DATA-MSG
21:                 else
22:                     - BROADCAST_DECISION(N, DATA-MSG)
23:                 end if
24:             end if
25:         end if
26:     end if
27: end if
28: procedure BROADCAST_DECISION(N, DATA-MSG)
29: if N < Cth
30:     - Broadcast DATA-MSG
31: else
32:     - Drop DATA-MSG
33: end if
34: end procedure
```

IV. Lecture d'information dans la matière communicante

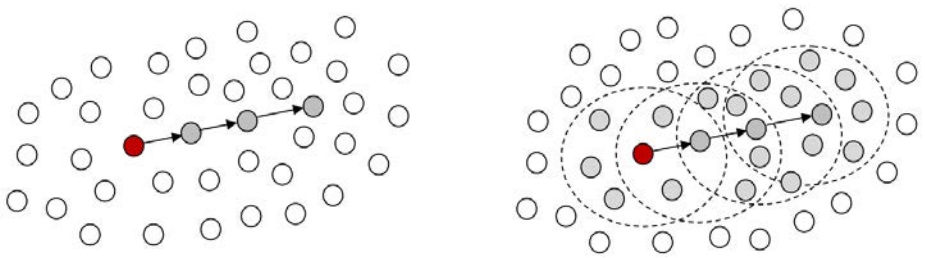
Comme décrit dans l'état de l'art, les protocoles de recherche d'informations existants dans la littérature sont basés sur l'inondation d'une requête dans le réseau. Ils collectent les valeurs de tous les nœuds et calculent le résultat au niveau du nœud de départ. Dans notre application sur la matière communicante, cela pourrait conduire à la congestion du réseau, à la surcharge du nœud maître et à la duplication des réponses puisque les informations sont uniformément répliquées dans un réseau très dense. Pour cette raison, nous avons choisi une solution de lecture d'informations basée sur le mécanisme de marche aléatoire (*Random Walk*) couplé à la technique *Pull Gossip* de découverte de voisinage à chaque saut.

IV.1. Random Walk et Pull Gossip

Le Random Walk, comme l'illustre la figure 38(a), consiste à transmettre un message requête d'un nœud à un de ses voisins aléatoirement choisi jusqu'à la fin de la longueur du chemin (le nombre de sauts que le message devrait effectuer) (Lima and Barros, 2007). Cependant, trouver l'information en utilisant le Random Walk peut ne pas être évident car le choix aléatoire des nœuds voisins peut conduire à un chemin sans résultat. Pour pallier cet inconvénient, nous proposons l'utilisation d'une technique de recherche dans le voisinage de chaque saut du chemin. Cette technique de découverte dans le voisinage peut être différenciée par son type de communication entre les nœuds voisins. Voici les trois principaux types de communication utilisés dans la littérature (Hakoura, 2011):

- Algorithmes à base de *Pull Gossip* : un nœud interroge ses voisins pour savoir s'ils ont de nouvelles informations. Le nœud ne recevra de nouvelles informations que si l'un de ses voisins en dispose d'au moins une.
- Algorithmes à base de *Push Gossip* : un nœud avec de nouvelles informations les envoie systématiquement à ses voisins.
- Algorithmes à base de *Pull-Push Gossip* : ces algorithmes présentent une combinaison des deux modèles décrits précédemment. Un nœud utilisant un tel algorithme envoie des informations à ses voisins quand il a des informations disponibles. Il demande également de nouvelles informations à ses voisins.

Nous nous sommes orientés vers la technique Pull Gossip. L'ajout des caractéristiques du Pull Gossip permettra d'améliorer la recherche aléatoire faite par le Random Walk par l'augmentation du nombre de nœuds couverts à chaque saut, ce qui augmente la chance de trouver l'information à chaque saut. L'exemple de la figure 40(a) montre que le nombre de nœuds explorés au cours d'une recherche d'information par un simple Random Walk est égal à 3. Par contre, la figure 40(b) montre que le nombre de nœuds couverts par la même recherche intégrant le Pull Gossip à chaque saut augmente le nombre de nœuds couverts à 20 nœuds pour le même nombre de sauts (3 sauts).



(a) Recherche par Random Walk (b) Recherche par Random Walk + Pull Gossip

Figure 40. Fonctionnement de Random Walk et Random Walk+Pull Gossip

De plus, une marche aléatoire combinée à un Pull Gossip permet de ne passer d'un nœud *X* à un autre *Y* que si l'information est absente dans le voisinage de *X*. À la réception de la requête

de recherche, le nœud diffuse un message de demande d'information dans son voisinage. Une fois cette phase établie, le nœud envoie la requête à un de ses voisins. Ce dernier continue lui-même la recherche en diffusant la requête à son propre voisinage et ainsi de suite jusqu'à trouver l'information. Le réseau doit mémoriser chaque saut pour acheminer le message de réponse vers la source de la requête (nœud maitre). Une fois l'information trouvée, la réponse est acheminée vers le nœud maitre en remontant le chemin parcouru par la requête.

L'ajout des caractéristiques du Pull Gossip permettra de contraindre la recherche aléatoire faite par le Random Walk. Il reste cependant des contraintes à prendre en considération, qui sont les suivantes :

- La recherche de l'information doit être limitée par une valeur seuil, car on ne doit pas chercher l'information jusqu'à l'infini dans le cas où le chemin ne conduirait à rien. Un seuil *TTL* (*Time To Live*) est défini. La valeur de *TTL* est décrémentée à chaque saut. Si *TTL* atteint la valeur zéro, le processus de recherche s'arrête.
- La recherche d'information doit être efficace et fiable. Cependant, en utilisant un simple Random Walk, le nœud choisit aléatoirement l'un de ses voisins, ce qui peut aboutir à des chemins aléatoires à faible qualité de service, à grand délai et à taux de pertes élevées. Pour cette raison, nous développons une technique de découverte de bordure de voisinage et une fonction d'évaluation de qualité de service à chaque saut. L'idée consiste à :
    - Découvrir les nœuds voisins distants comme l'illustre la figure 41 (qui sont définis comme nœuds de bordure du voisinage). La proposition de cette technique a été faite pour couvrir plus de nouveaux nœuds dans le prochain saut dans le processus de *Random Walk+Pull Gossip*. Au contraire, si on ne choisit pas des nœuds qui sont dans la bordure du voisinage (des nœuds proches du nœud source), on risque de couvrir un faible nombre de nouveaux nœuds au prochain saut dans le processus de recherche. Par exemple sur la figure 42(a), le nœud du prochain saut est choisi aléatoirement. Le nombre des nouveaux nœuds couverts lors de la recherche dans le voisinage de prochain saut est égal à 6 (nœuds en noir sur la figure 42(a)). Alors qu'avec le choix d'un nœud sur la bordure de voisinage comme prochain saut, le nombre des nœuds couverts est égal à 12 (nœuds en noir sur la figure 40(b)). Cela permet ainsi d'augmenter la chance de trouver l'information à chaque nouveau saut.
    - Parmi les voisins distants, le nœud choisit le voisin le plus puissant (par exemple le plus haut niveau d'énergie restante, ou le plus faible taux d'interférence). Le choix du nœud, pour le prochain saut, le plus puissant nous permet d'acheminer la requête le long d'un chemin ayant une haute qualité de service.

Dans la suite, nous présentons point par point notre protocole de recherche d'information dans la matière communicante nommé RaWPG (Random Walk and Pull Gossip data retrieval protocol).

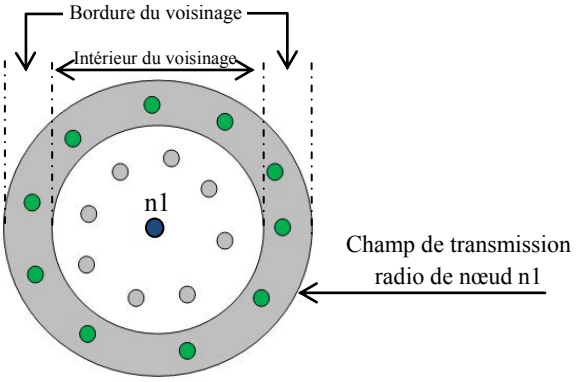
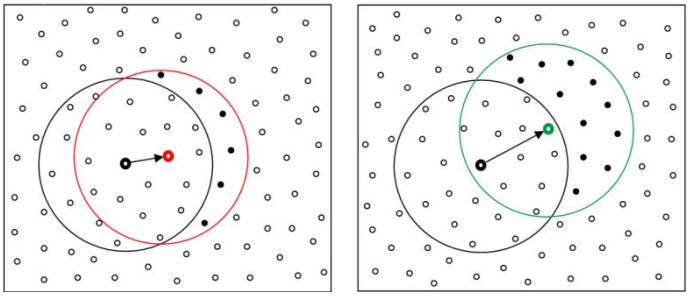


Figure 41. Ensemble des nœuds de bordure dans un voisinage de nœud de capteurs



(a) (b)

Figure 42. Effet du choix des noeuds voisins distants

IV.2. Algorithme de sélection des nœuds de bordure du voisinage

Cet algorithme représente une phase primordiale pour le fonctionnement final du protocole RaWPG. C'est une phase qui s'exécute au début de la vie de la matière communicante (auto-organisation du réseau dans la matière) pour que ce dernier soit capable d'exécuter le protocole de recherche d'informations RaWPG tout au long de son cycle de vie. En effet, pour choisir les nœuds les plus éloignés parmi une liste des nœuds voisins, la solution habituelle consiste à estimer la distance entre le nœud et tous ses voisins. Dans ce qui suit, sont présentées les techniques d'estimation de distance existantes pour les réseaux de capteurs sans fil. Ces techniques sont ensuite analysées dans le contexte de préfabriqué béton.

IV.2.1. Techniques d'estimation de distance dans les réseaux de capteurs

Deux techniques d'estimation de distance existent dans la littérature : les techniques basées sur *la propagation du signal radio* et les techniques basées sur *la cardinalité d'intersection de voisinage.*

a) Techniques basées sur la propagation du signal radio

L'estimation de distance (ranging) basée sur la propagation du signal radio peut être réalisée en utilisant des informations comme le temps d'arrivée (ToA), la différence de temps d'arrivée (TDoA), l'angle d'arrivée (AoA) et l'indicateur de la force de signal reçu (RSS). Les différentes techniques sont présentées ci-après :

- Time of Arrival (ToA) : La distance d_{ij} entre un nœud émetteur *i* et un nœud récepteur *j* peut être déterminée en utilisant le temps de propagation *t* et la vitesse d'un signal *v* (Mao et al., 2007). Il y a deux types d'estimation :

Aller-simple (One-way) : elle représente la propagation du signal avec une précision d'horloge importante à partir d'un émetteur *i* jusqu'à un récepteur *j*, et qui s'exprime par la relation : $d_{ij} = (t_1 - t_2)v$. La Figure 43 partie (a) montre graphiquement que le temps t_1 correspond à l'instant de sortie d'un signal de l'émetteur, et le temps t_2 à l'instant de réception du récepteur.

Aller-retour (Two-ways) : elle utilise un aller-retour du signal de propagation. La Figure 43 partie (b) montre les nouveaux temps de propagation t_3 et t_4 qui correspondent aux instants de sortie et de réception du signal de retour, qui permettent de déterminer la distance selon la relation : $d_{ij} = (t_4 - t_1)(t_3 - t_2)v$

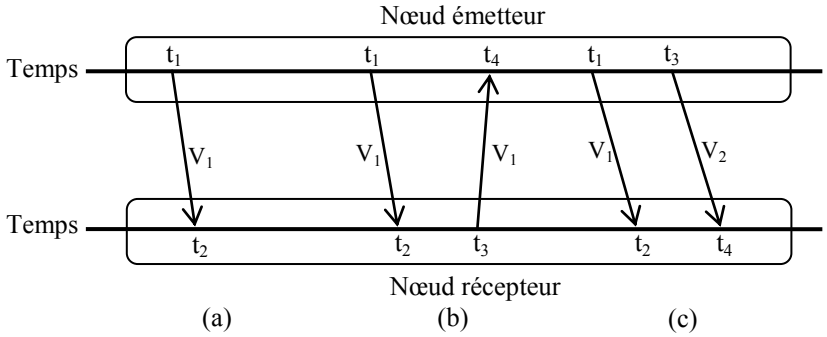


Figure 43. Techniques d'estimation de distance : ToA et TDoA

- Time Difference of Arrival (TDoA) : Cette procédure utilise deux signaux qui se propagent avec des vitesses différentes v_1 et v_2 (par exemple un signal radio et un autre de bruit). Le signal v_1 est envoyé par l'émetteur *i* à l'instant t_1, et est reçu par le récepteur *j* à l'instant t_2. Le signal v_2 est envoyé à un moment t_3 par l'émetteur plus un temps d'attente t_{wait}, et est reçu à l'instant t_4 par le récepteur (voir figure 43 partie (c)). La distance entre l'émetteur et le récepteur est alors déterminée par la relation : $d_{ij} = (v_1 - v_2)(t_4 - t_2 - t_{wait})$

Cette technique ne nécessite pas de synchronisation d'horloge, et peut avoir une grande précision, mais nécessite aussi un équipement supplémentaire.

- Angle of Arrival (AoA) : Aussi appelée DoA (Direction of Arrival). Cette méthode détermine la position d'un point dans l'espace à partir d'un réseau d'antennes avec des signaux de propagation dans différentes directions. Dans la littérature (Peng and Sichitiu, 2006), il y a des méthodes disponibles pour déterminer la position d'un ou plusieurs objets, et donc d'estimer la distance relative entre eux. Le calcul vient de l'utilisation des propriétés géométriques qui permettent d'utiliser la connaissance de la valeur des angles, afin de déterminer la distance entre les nœuds.

- Received Signal Strength (RSS) : La force du signal diminue avec la distance, et peut être estimée par l'indicateur *RSSI* (Received Signal Strength Indicator) (Ash and Potter, 2004). Ensuite, il est possible de déterminer la distance entre un émetteur *i* et un récepteur *j* avec la relation suivante : $d_{ij} = \sqrt[\alpha]{c \times \frac{P_i}{P_j}}$

Dans ce cas, *c* est le gain d'antenne ou l'amplification de la puissance d'une antenne, *P* la puissance du signal reçu, et α le coefficient de perte de puissance dans le modèle de propagation.

b) Techniques basées sur la cardinalité d'intersection de voisinage

Cette catégorie de techniques d'estimation de distance entre les nœuds de capteurs est basée sur la connectivité dans le voisinage. Ces techniques sont moins précises que celles précédemment présentées, mais ne nécessitent pas de matériels supplémentaires pour les nœuds.

Etant donné deux nœuds *n1* et *n2* distants de *d* ($d \leq r$) et deux disques ayant un même rayon *r* de communication radio (voir figure 44). Deux algorithmes principaux existent dans la littérature : NIDES et MLE.

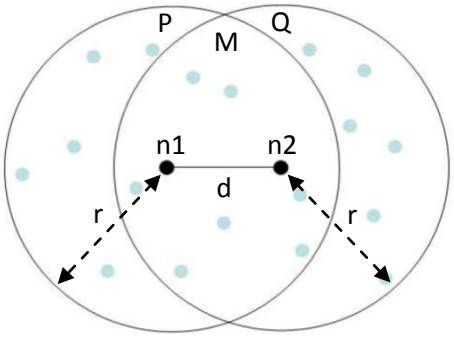


Figure 44. Intersection des cercles de communication radio de deux nœuds voisins n1 et n2

- NIDES : Les travaux de (Buschmann et al., 2006) et (Buschmann et al., 2007) présentent une technique d'estimation de distance appelée NIDES (Neighborhood Intersection Distance Estimation Scheme) basée sur la cardinalité d'intersection des voisinages. Cette technique est bien appliquée dans les réseaux de capteurs sans fil avec un déploiement uniforme ou aléatoire des nœuds. L'approche ne nécessite pas de matériels spécifiques ou des mesures de propriétés physiques du signal radio propagé. Au lieu de cela, elle estime la distance en se basant sur le calcul de la cardinalité des ensembles d'intersection de voisinages des nœuds adjacents (les nœuds qui peuvent communiquer directement, appelés aussi dans ce rapport « nœuds voisins »). La technique ne s'applique pas pour des nœuds qui ne peuvent pas communiquer directement entre eux.

La distance est estimée par la fonction ci-dessous. Elle prend en entrée la cardinalité du voisinage de *n1* (le nœud qui estime la distance) et la cardinalité d'intersection des ensembles des nœuds voisins de *n1* et *n2*, et déduit la distance estimée à l'aide de la relation suivante :

$$d = -4\sqrt{2}\, r\ \sin\left(\frac{\arcsin\left(\frac{3\sqrt{2}(4\,A - \pi r^2)}{16\, r^2}\right)}{3}\right)$$

Avec : *A* est le quart de la zone d'intersection des deux cercles de communication de *n1* et *n2*.

A est défini par la formule suivante : $A = 0.25\frac{s}{n}\pi r^2$

n est le nombre des nœuds voisins de *n1*.

s est le nombre de nœuds en commun (la cardinalité d'intersection) entre *n1* et *n2*.

- MLE : MLE (Maximum-Likelihood Estimator) est une autre solution proposée par (Huang et al., 2014) qui a démontré sa précision dans un réseau de capteurs statiques avec un déploiement uniforme ou aléatoire des nœuds. Comme $d \leq r$, les deux cercles de communication radio se croisent et créent trois régions disjointes. Une formule représentant la surface de la zone d'intersection entre les nœuds *n1* et *n2* a été proposée :

$$f(d) = \frac{2S}{\pi} arccos\left(\frac{d}{2r}\right) - d\sqrt{r^2 - \frac{d^2}{4}}$$

Avec : $S = \pi\, r^2$

L'intersection entre *n1* et *n2* donne trois zones comme le montre la figure 42. Les nœuds résidant dans la zone au milieu *M* sont les nœuds voisins en commun entre *n1* et *n2*. Les nœuds résidant dans la gauche (zone *P*) sont les nœuds voisins de *n1*, n'appartenant pas au voisinage de *n2*. De même, les nœuds résidant dans la droite (zone *Q*) sont les nœuds voisins de *n2* non-commun avec *n1*. *M*, *P* et *Q* définissent trois variables aléatoires indépendantes qui suivent la loi de Poisson de moyenne $\lambda f(d)$, $\lambda(S-f(d))$ et $\lambda(S-f(d))$, respectivement. Les valeurs de *M*, *P* et *Q* sont calculées et obtenues après un échange d'informations concernant les voisinages de *n1* et *n2*. La distance *d* entre *n1* et *n2* est estimée en utilisant la formule suivant :

$$\hat{d} = \begin{cases} f^{-1}(S), & si\ M = P = Q = 0 \\ f^{-1}(\hat{\rho} S) & sinon \end{cases}$$

Avec : $\hat{\rho} = \frac{2M}{2M+P+Q}$

c) Analyse et critères de choix

Dans la technique RSS, différents modèles sont utilisés pour traduire la force du signal en une distance. Cette solution n'est efficace que si les modèles de canaux sont précis, c'est le cas dans un environnement contrôlé comme dans un laboratoire. Son utilisation dans la matière communicante ne semble pas réaliste. En effet dans certains matériaux, il faudrait considérer des problèmes comme le *multi-path fading* et la propagation de signaux irréguliers (Tran-Dang et al., 2014).

Bien que les méthodes ToA, TDoA et AoA puissent fournir des estimations de distance avec une précision très élevée, ces techniques s'appuient également sur des matériels supplémentaires avec un taux de consommation d'énergie plus important (Mao et al., 2007). Ces méthodes sont donc moins intéressantes pour des micro-nœuds de capteurs à faible puissance dans les matériaux communicants.

Par rapport à ces contraintes matérielles et conceptuelles, nous nous sommes orientés vers les techniques s'appuyant sur la connectivité du voisinage et la cardinalité d'intersection entre les deux cercles de communication radio pour deux nœuds adjacents *n1* et *n2* dans la matière.

En outre, dans notre contexte, nous ne recherchons pas précisément une distance mais plutôt trouver les nœuds les plus éloignés parmi un ensemble de voisins. Cette alternative fait d'ailleurs usage de la propriété de diffusion et de connectivité radio, disponible pour tout réseau de capteurs sans fil, ne nécessitant pas de valeurs connues à l'avance comme le rayon de champ de communication radio *r*.

Dans la suite de ce document, le fonctionnement de notre algorithme pour la sélection des nœuds voisins distants est détaillé.

IV.2.2. Algorithme de sélection des nœuds voisins distants

Dans le contexte de la matière communicante, les nœuds de capteurs sont déployés uniformément dans la matière. Le fonctionnement de notre solution se base sur le fait que les nœuds situés l'un à côté de l'autre partagent beaucoup de nœuds voisins, tandis que les nœuds éloignés partagent seulement quelques voisins comme l'illustre la figure 45. Par exemple dans la figure 45(a), les nœuds *n1* et *n2* sont très proches et la cardinalité de la zone d'intersection *M* est égale à 11. Par contre dans la figure 45(b), le nœud *n2* se trouve à la bordure de zone de communication de *n1* et la cardinalité de zone d'intersection est égale à 7. Donc, l'augmentation de la distance entre *n1* et *n2* diminue la cardinalité de zone d'intersection *M* sachant que *n1* et *n2* sont toujours en communication directe (nœuds voisins).

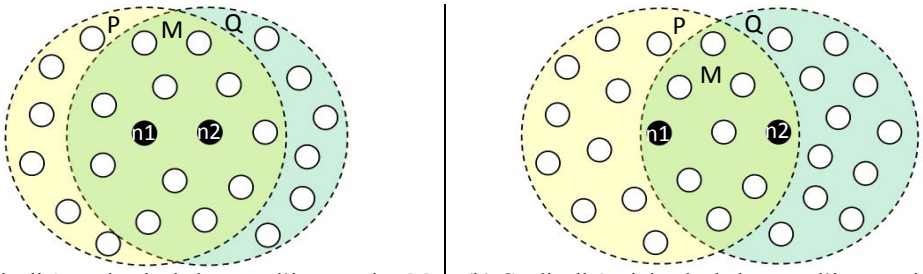
Désignons par V_{n1} et V_{n2} comme étant respectivement les ensembles de nœuds voisins de *n1* et de *n2* avec $n1 \in V_{n2}$ et $n2 \in V_{n1}$.

L'idée consiste à trouver l'ensemble *A* des nœuds *n2* qui ont le nombre minimal d'éléments communs entre V_{n2} et V_{n1}. Cet ensemble est constitué des nœuds $n2 \in V_{n1}$ tel que $card(V_{n1} \cap V_{n2})$ soit minimal. Cela est présenté par la formule suivante :

$$A=\{n2 \in V_{n1} \text{ tel que } \forall\, x \in V_{n1} \text{ et } x \neq n2,\ card(V_{n1} \cap V_{n2}) \leq card(V_{n1} \cap V_x)\}$$

Avec : *card (V)* le nombre d'éléments contenu dans un ensemble *V*.

En utilisant cette formule, chaque nœud dans la matière peut trouver l'ensemble de ses voisins qui sont les plus distants (sur la bordure de voisinage). Dans le reste de cette section, on présente les entêtes des messages utilisés et l'algorithme de sélection des nœuds voisins distants.



(a) Cardinalité maximale de la zone d'intersection M (b) Cardinalité minimale de la zone d'intersection M

Figure 45. Variation de cardinalité de la zone d'intersection M suivant la distance entre n1 et n2

a) Entêtes des messages d'algorithme de sélection des nœuds voisins distants

Pour la sélection des nœuds voisins distants, trois types de messages ont été employés :

- **HELLO** : C'est un message qui permet à chaque nœud d'annoncer son identifiant à tous ses voisins. La structure de ce message est présentée dans la figure 46.

Type	ID_SOURCE

Figure 46. Structure du message HELLO

Avec :

Type : est utilisé pour connaitre le type de message reçu.

ID_SOURCE : contient l'identifiant de l'émetteur du message HELLO.

- **TAB_EXCHANGE**: C'est un message qui permet à chaque nœud d'annoncer sa table de voisinage à tous ses voisins. La structure de ce message est présentée dans la figure 47.

Type	ID_SOURCE	N_{Table}

Figure 47. Structure du message TAB_EXCHANGE

Avec :

Type : est utilisé pour connaitre le type de message reçu.

ID_SOURCE : contient l'identifiant de l'émetteur du message TAB_EXCHANGE.

N_{Table} : contient la table de voisinage de l'émetteur du message TAB_EXCHANGE.

- **FAR_NEIGHBORS**: C'est un message qui permet à chaque nœud d'annoncer dans son voisinage la liste de ses nœuds voisins distants. La structure de ce message est présentée dans la figure 48.

Type	ID_SOURCE	$L_{Farthest}$

Figure 48. Structure du message FAR_NEIGHBORS

Avec :

Type : est utilisé pour connaitre le type de message reçu.

ID_SOURCE : contient l'identifiant de l'émetteur du message FAR_NEIGHBORS.

$L_{Farthest}$: contient la liste des nœuds voisins distants du nœud émetteur du message FAR_NEIGHBORS.

b) Fonctionnement de l'algorithme de sélection des nœuds voisins distants

La sélection des nœuds voisins distants est présentée par l'*Algorithme2*. Soit *X* un nœud de capteurs dans la matière communicante et N_{Table} sa table de voisinage. L'algorithme est basé sur la diffusion comme modèle de communication. Pour découvrir les nœuds de bordure dans chaque voisinage, chaque nœud dans la matière annonce son identifiant vers tous ses voisins en utilisant le message HELLO (ligne 1). Puis, le nœud se met en écoute de messages HELLO émis par ses voisins (ligne 2). Pour chaque message HELLO reçu, le nœud *X* ajoute l'identifiant de l'émetteur dans sa table de voisinage N_{Table} (lignes 3-5). Ce dernier sera ensuite inséré dans un message TAB_EXCHANGE et diffusé vers tous les voisins (ligne 6). Ensuite, le nœud se met de nouveau en écoute des messages TAB_EXCHANGE de ses voisins (ligne 7). Pour chaque message TAB_EXCHANGE reçu, le nœud *X* calcule la cardinalité des nœuds en commun entre la table N_{Table} de *X* et la table N_{Table} transportées par le message TAB_EXCHANGE reçu (lignes 8-10). Un nœud *Y* est considéré sur la bordure de voisinage du nœud *X*, si l'intersection de N_{Table} de *X* et N_{Table} de *Y* a une cardinalité minimale par rapport aux autres nœuds voisins de *X*. Le nœud *X* ajoute dans $L_{Farthest}$ tous ses voisins qui ont une cardinalité minimale (ligne 11). Ensuite, le nœud *X* insère $L_{Farthest}$ dans un message FAR_NEIGHBORS et le diffuse dans son voisinage et il se met en écoute de l'émission d'un message identique de ses voisins (lignes 12 et 13). Lors de la réception d'un message FAR_NEIGHBORS, le nœud *X* vérifie si son identifiant existe dans la liste $L_{Farthest}$ portée par ce message. Si l'identifiant de *X* existe, le nœud *X* s'auto-configure comme un nœud de bordure pour le nœud d'identifiant ID_SOURCE de l'émetteur de FAR_NEIGHBORS (lignes 14-20).

Une fois que la phase d'identification des nœuds de bordure de voisinage de chaque nœud dans la matière est effectuée, le réseau de capteurs peut commencer l'exécution du protocole de recherche d'information RaWPG.

Algorithme2: Sélection des nœuds voisins les plus distants

For any node *X, do:*

1: - Broadcast *HELLO* to all neighbor nodes
2: - Wait for reception of *HELLO* messages from neighbors
3: **for** each received *HELLO* **do**
4: - Add *ID_SOURCE* in N_{Table}
5: **end for**
6: - Broadcast *TAB_EXCHANGE* to all neighbor nodes
7: - Wait for reception of *TAB_EXCHANGE* messages from neighbors
8: **for** each received *TAB_EXCHANGE* **do**
9: - Calculate the cardinality of intersection between $N_{Table\text{-}X}$ of *X* and $N_{Table\text{-}Neighbor}$ of *TAB_EXCHANGE* : card($N_{Table\text{-}X} \cap N_{Table\text{-}Neighbor}$)
10: **end for**
11: - Select the neighbors who have the minimum intersection cardinality : min(card($N_{Table\text{-}X} \cap N_{Table\text{-}Neighbor}$)

```
- Broadcast FAR_NEIGHBORS
- Wait for reception of FAR_NEIGHBORS messages from neighbors
for each received FAR_NEIGHBORS do
    if ID of X is in the list L_Farthest of the message
        - Configured as farthest neighbor for ID_SOURCE of the message
    else
        - Drop FAR_NEIGHBORS
    end if
end for
```

IV.3. Protocole de recherche d'information RaWPG

Dans le reste de ce chapitre, on présente d'abord une description générale du fonctionnement du protocole RaWPG, et on définit la fonction de coût de liaison pour la sélection du meilleur prochain saut en terme de qualité de service. Ensuite, la structure des messages est décrite. Enfin, l'algorithme du protocole est détaillé.

IV.3.1. Fonctionnement du protocole RaWPG

Afin de chercher une information dans la matière, l'utilisateur se connecte sans fil à un nœud maitre et lui transmet le message requête. Le nœud maitre commence à rechercher l'information dans son voisinage en diffusant une requête à tous ses nœuds voisins comme le montre la figure 49(a) (le nœud rouge représente le maître R, et le cercle gris représente le champ de transmission radio de R). Si un des nœuds voisins envoie une réponse, le processus de recherche s'arrête. Sinon, les nœuds configurés comme nœud de bordure de voisinage pour le maitre R répondent par des messages contenant leur énergie résiduelle, leur taille du tampon libre (taille de tampon = taille de la file d'attente de réception) et leur qualité de liaison SNR comme le montre la figure 49(b). Dans cette figure, chaque nœud vert représente un nœud de bordure de voisinage. Une fonction de coût de liaison est adoptée pour sélectionner le nœud voisin distant le plus puissant comme prochain saut de la requête. Le choix d'un nœud voisin puissant parmi les nœuds de bordure de voisinage permet l'acheminement de la requête par un chemin de haute qualité de service. Soit Y le prochain saut sélectionné par R. Le nœud maitre R transmet la requête au nœud Y (figure 49(c)).

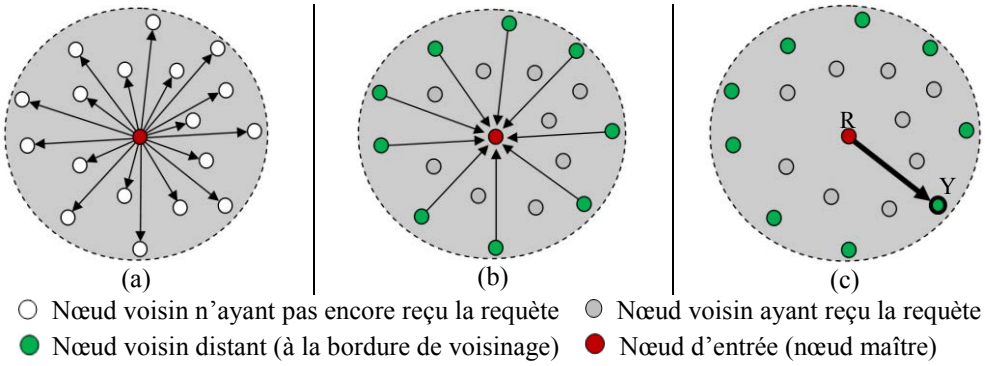


Figure 49. Recherche d'information initiée par le nœud maitre

De même que le nœud R, le nœud Y diffuse la requête vers ses nœuds voisins. Seuls les nœuds voisins de Y n'ayant pas déjà reçu la requête répondent, les autres nœuds voisins communs avec le nœud maitre R étant passés en mode inactif.

Désignons El_Y l'ensemble des nœuds voisins de Y qui n'ont pas encore reçu la requête comme le montre la figure 50(a). El_Y est défini par l'ensemble des nœuds voisins de Y non commun avec R.

Si un nœud de l'ensemble El_Y a l'information demandée dans sa mémoire, il répond par un message réponse qui contient cette information. Dans ce cas, le nœud Y transmet ce message d'information vers la source de la requête dans son voisinage qui est R, et le processus de recherche s'arrête. Cependant, si aucun nœud dans El_Y n'a l'information dans sa mémoire, seuls les nœuds de bordure du voisinage de Y appartenant à El_Y vont répondre chacun par un message contenant leur énergie résiduelle, leur taille de tampon libre et leur qualité de liaison SNR (voir figure 50(b)). Lors de la réception de ces messages, le nœud Y choisit le meilleur voisin distant dans El_Y en utilisant la fonction de coût de liaison. Soit Z le prochain saut sélectionné par Y. Le nœud Y transmet la requête au nœud Z (voir figure 50(c)).

Ce processus continue de la même manière au niveau du nœud Z dans son ensemble El_Z défini par l'ensemble des nœuds voisins de Z non commun avec Y pour trouver un nœud de prochain saut sur le chemin de recherche.

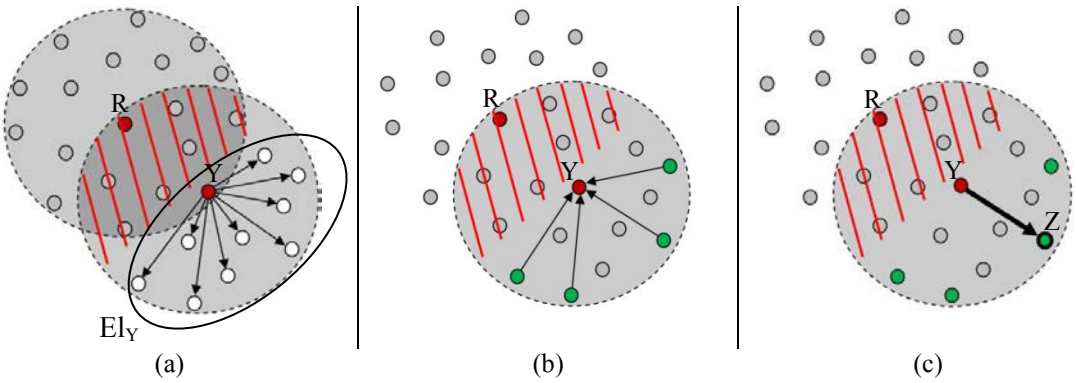


Figure 50. Recherche d'information au niveau de noeud Y

Ce processus se poursuit jusqu'à trouver l'information dans le voisinage d'un saut dans le chemin de recherche comme le montre la figure 51. Une fois trouvée, l'information remonte vers le nœud maitre par le chemin inverse parcouru par la requête (voir la figure 51). Le processus de recherche s'arrête aussi quand le compteur TTL devient zéro. Dans ce cas, un message d'échec remonte vers le nœud maitre sur le chemin inverse parcouru par la requête.

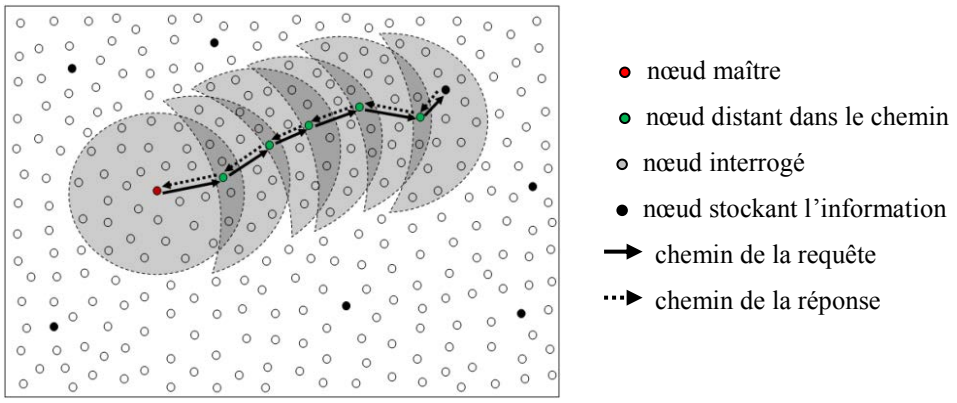


Figure 51. Acheminement de requêtes et réponses dans le réseau de capteurs par RaWPG

IV.3.2. Fonction de coût de liaison

Le coût de liaison est utilisé pour choisir le nœud voisin distant qui offre la meilleure qualité de service. Nous utilisons une fonction de coût similaire à (Ben-Othman and Yahya, 2010). Notre fonction de coût comprend un facteur d'énergie, un facteur de mémoire tampon et un facteur d'interférence, avec des poids appropriés notés α, β et γ. On considère un nœud X dans le chemin de recherche. Le nœud X choisit, comme prochain saut de requête, le voisin distant Y qui maximise la valeur de la fonction suivante :

$$Prochain_{Saut} = \max_{Y \in L_{Fart\ hest}} \{\alpha E_{resd\ ,Y} + \beta B_{buffer\ ,Y} + \gamma I_{interference\ \ ,XY}\}$$

Avec :

$E_{resd\ ,Y}$: l'énergie restante de nœud Y.

$B_{buffer\ ,Y}$: la taille du mémoire tampon du nœud Y.

$I_{interference\ \ ,XY}$: le rapport signal sur bruit (SNR) du lien radio entre un nœud X et Y.

Cette formule permet d'avoir un compromis entre ces trois facteurs. En outre, les poids des facteurs α, β et γ ajustent les priorités dans l'évaluation du coût de liaison. Un grand poids donne plus de priorité au facteur associé pour l'évaluation du choix de la qualité et l'importance du nœud de prochain saut. La sélection du nœud qui offre la meilleure qualité de service, doit assurer une fiabilité de transfert de la requête du nœud X au nœud Y et du retour pour la réponse du nœud Y vers X.

IV.3.3. Entêtes des messages de RaWPG

Le protocole RaWPG utilise cinq messages pour accomplir le processus de recherche d'une information dans la matière communicante : REQUEST, REQUEST_NG, FAR_INFO, RESPONSE et FAILURE.

- **REQUEST**: C'est un message requête qui contient la référence de l'information demandée par l'utilisateur. La structure de ce message est présentée dans la figure 52.

TYPE	ID_SOURCE	ID_DESTINATION	TIMESTAMP	REF_DATA	TTL

Figure 52. Structure du message REQUEST

Avec :

TYPE : est utilisé pour connaitre le type de message reçu.

ID_SOURCE : contient l'identifiant de l'émetteur du message *REQUEST*.

ID_DESTINATION : contient l'identifiant du destinataire (nœud du prochain saut) du message *REQUEST*.

TIMESTAMP : c'est l'identifiant de la phase de lecture. Il indique l'instant de transmission de REQUEST dans le réseau par le nœud maitre.

REF_DATA : c'est la référence de l'information demandée.

TTL : c'est le seuil du nombre maximal de sauts autorisés pour l'acheminement du message *REQUEST* dans le réseau.

- REQUEST_NG : C'est un message requête de recherche dans le voisinage de chaque saut. Il permet de demander si l'information existe dans les nœuds voisins. REQUEST_NG contient la même référence que celle portée par REQUEST. La structure de ce message est présentée dans la figure 53.

TYPE	ID_SOURCE	TIMESTAMP	REF_DATA

Figure 53. Structure du message REQUEST_NG

Avec :

TYPE : est utilisé pour connaitre le type du message reçu.

ID_SOURCE : contient l'identifiant de l'émetteur du message *REQUEST_NG*.

TIMESTAMP : porte la même valeur que celle contenue dans *TIMESTAMP* du *REQUEST*. Ce champ est utilisé pour identifier *REQUEST_NG* et pour éviter aux voisins communs de deux sauts successifs de traiter plusieurs fois ce message.

REF_DATA : c'est la référence de l'information demandée.

- FAR_INFO : C'est le message envoyé par un nœud de bordure de voisinage vers le nœud source de REQUEST_NG. Ce message contient les informations nécessaires pour choisir le nœud de bordure le plus adapté suivant la fonction du coût de liaison. La structure de ce message est présentée dans la figure 54.

TYPE	ID_SOURCE	ID_DESTINATION	ENERGY	BUFFER	SNR

Figure 54. Structure du message FAR_INFO

Avec :

TYPE : est utilisé pour connaitre le type de message reçu.

ID_SOURCE : contient l'identifiant de l'émetteur du message *FAR_INFO*.

ID_DESTINATION : contient l'identifiant du destinataire du message *FAR_INFO*. Le destinataire contient exactement la même valeur que la valeur du champ ID_SOURCE du message REQUEST_NG reçu.

ENERGIE_RESTANTE : c'est le niveau d'énergie restant dans le nœud émetteur du message *FAR_INFO*.

BUFFER : c'est la taille de mémoire tampon libre dans le nœud émetteur du message *FAR_INFO*.

SNR : c'est le rapport SNR de la liaison radio entre le nœud émetteur de REQUEST et le nœud émetteur de FAR_INFO.

- **RESPONSE :** C'est le message de réponse envoyé par le nœud qui a l'information demandée dans sa mémoire. Cette information est référencée par REF_DATA dans le message REQUEST et REQUEST_NG. La structure de ce message est présentée dans la figure 55.

TYPE	ID_SOURCE	ID_DESTINATION	INFORMATION

Figure 55. Structure du message RESPONSE

Avec :

TYPE : est utilisé pour connaitre le type du message reçu.

ID_SOURCE : contient l'identifiant de l'émetteur du message *RESPONSE*.

ID_DESTINATION : contient l'identifiant du destinataire du message *RESPONSE*.

INFORMATION : contient l'information demandée par l'utilisateur.

- **FAILURE :** C'est le message envoyé par le nœud qui reçoit le message REQUEST avec TTL=0 dans le processus de recherche. FAILURE est renvoyé vers le nœud maitre à travers le même chemin parcouru par le message REQUEST. La structure de ce message est présentée dans la figure 56.

TYPE	ID_SOURCE	ID_DESTINATION

Figure 56. Structure du message FAILURE

Avec :

TYPE : est utilisé pour connaitre le type du message reçu.

ID_SOURCE : contient l'identifiant de l'émetteur du message *FAILURE*.

ID_DESTINATION : contient l'identifiant du destinataire du message *FAILURE*.

IV.3.4. Algorithme de lecture RaWPG

La description détaillée de l'algorithme de recherche RaWPG comporte trois traitements différents : réception d'un message REQUEST, réception d'un message REQUEST_NG et réception d'un message RESPONSE ou FAILURE. Dans ce qui suit, on détaille le traitement de chaque message par un nœud de capteurs dans la matière.

a) Traitement du message REQUEST

L'algorithme de traitement du message REQUEST est donné par l'*Algorithme3*. Le message REQUEST est acheminé sur un chemin de différents nœuds dans la matière à partir du nœud maitre jusqu'à un nœud qui contient l'information ou jusqu'à la fin du compteur TTL. Si l'un de ces deux cas arrive, le message RESPONSE ou FAILURE est acheminé sur le même chemin parcouru par REQUEST mais dans le sens inverse pour atteindre le nœud maitre. Pour cette raison, lors de la réception du message REQUEST, le nœud enregistre

temporairement l'identifiant du nœud émetteur ID_SOURCE de REQUEST pour construire le chemin inverse vers le maitre (ligne 1). D'où, quand un nœud reçoit un message RESPONSE ou FAILURE, il connaît l'identifiant du prochain destinataire pour acheminer ces messages vers le maitre.

Ensuite, le nœud vérifie s'il a l'information demandée dans sa mémoire. Dans ce cas, il l'envoie dans un message RESPONSE vers ID_SOURCE de REQUEST (lignes 2 et 3). Sinon, le nœud vérifie que le champ TTL dans la requête n'est pas égal à zéro. Un message FAILURE est retourné vers le maitre si c'est le cas (lignes 4-6). Sinon, le nœud commence la découverte de l'information dans son voisinage. Pour cela, un message REQUEST_NG est créé et diffusé vers tous les nœuds voisins, puis le nœud se met en attente pour la réception possible d'un message RESPONSE. De plus, un voisin configuré comme bordure pour le nœud source de REQUEST_NG, répond par un message FAR_INFO qui contient le niveau de son énergie restante, sa taille du tampon libre et son SNR (lignes 7-9). Durant la période d'attente, si un RESPONSE est reçu, le nœud achemine ce message à son nœud précédent sur le chemin de recherche, et le processus de recherche s'arrête (lignes 12 et 13). Si aucun message RESPONSE n'est reçu, cela signifie que l'information n'existe pas dans le voisinage. Dans ce cas, le nœud cherche le nœud de bordure le plus adapté en se basant sur les messages FAR_INFO reçus (lignes 13 et 14). Ensuite, le nœud décrémente TTL et transmet le message REQUEST vers le meilleur nœud de bordure trouvé (lignes 14 et 15).

b) Traitement du message REQUEST_NG

L'algorithme de traitement du message REQUEST_NG est donné par l'*Algorithme4*. Lors de la réception d'un message REQUEST, le nœud diffuse dans son voisinage un message REQUEST_NG pour interroger ses nœuds voisins. Quand un nœud voisin reçoit un message REQUEST_NG, il vérifie tout d'abord qu'il n'a pas précédemment reçu le même message pour la même opération de recherche. Cela évite d'interroger plusieurs fois les nœuds communs des sauts de chemin de recherche. L'identification de REQUEST_NG est faite par le couple REF_DATA / TIMESTAMP, qui est un champ fixé par le nœud maitre tout au long d'une opération de recherche. Si un nœud reçoit un message avec un identifiant déjà traité pendant la période de recherche, il le supprime (lignes 1 et 2). Cependant, si le nœud reçoit REQUEST_NG pour la première fois, il vérifie l'existence d'information dans sa mémoire. Si l'information existe, le nœud envoie RESPONSE au nœud source de REQUEST_NG (lignes 3-5). Sinon, si le nœud est configuré en nœud de bordure de voisinage pour le nœud source de REQUEST_NG, le nœud envoie un message FAR_INFO qui contient son niveau d'énergie restante, sa taille de tampon libre et son SNR de liaison radio avec la source REQUEST_NG (lignes 6-8).

Si le nœud n'a ni l'information et n'est ni configuré comme nœud de bordure, il supprime le message et se met en mode inactif (lignes 9 et 10).

Algorithme3: Traitement du message REQUEST

For a node *X*: upon reception of a *REQUEST* message, do:

1: - Store *ID_SOURCE* of *REQUEST*
2: **if** *REF_DATA* exists in the memory of *X*
3: - Send back *RESPONSE* message to *ID_SOURCE* of *REQUEST*
4: **else**
5: **if** *TTL*=0
6: - Send back *FAILURE* message to *ID_SOURCE* of *REQUEST*
7: **else**
8: - Broadcast *REQUEST_NG*
9: - Wait for reception of *FAR_INFO* of the farthest neighbors and possible *RESPONSE* message
10: **if** *REPONSE* received
11: - Send back *REPONSE* to *ID_SOURCE* of *REQUEST*
12: **else**
13: - Compute $max\{\alpha E_{resd,Y} + \beta B_{buffer,Y} + \beta I_{interference,XY}\}$ using the received *FAR_INFO* messages
14: - Decrement *TTL* (*TTL*←*TTL*-1)
15: - Send *REQUEST* to the best farthest neighbor
16: **end if**
17: **end if**
18: **end if**

Algorithme4: Traitement du message REQUEST_NG

For a node *X*: upon reception of a *REQUEST_NG* message, do:

1: **if** *REQUEST_NG* is already received, *REF_DATA / TIMESTAMP* is known
2: - Drop *REQUEST_NG*
3: **else**
4: **if** *REF_DATA* exists in the memory of *X*
5: - Send back *REPONSE* message to *ID_SOURCE* of *REQUEST_NG*
6: **else**
7: **if** *X* if configured as a farthest neighbor for *ID_SOURCE* of *REQUEST_NG*
8: - Send back *FAR_INFO* message to *ID_SOURCE* of *REQUEST_NG*
9: **else**
10: - Drop *REQUEST_NG*
11: **end if**
12: **end if**
13: **end if**

c) Traitement du message RESPONSE ou FAILURE

L'acheminement des messages RESPONSE ou FAILURE se fait sur le chemin inverse de celui de REQUEST. Donc, lors de la réception de ce type de message, le nœud l'envoie directement à son nœud précédent dans le chemin de transmission de REQUEST. En effet, c'est le nœud qui a l'adresse enregistrée lors de la réception de REQUEST (ligne 1 de l'*Algorithme3* de traitement de message REQUEST).

V. Conclusion

Ce chapitre a présenté les protocoles proposés : le protocole de dissémination USEE pour un stockage uniforme d'informations dans la matière, et le protocole de recherche d'informations RaWPG adapté à USEE. Ainsi, le comportement des nœuds dans la matière, les entêtes protocolaires des messages échangés et les algorithmes ont été détaillés.

Ces protocoles proposés doivent être maintenant validés par simulation dans un environnement spécifique aux réseaux de capteurs sans fil. Dans le chapitre suivant nous allons commencer par la description du travail que nous avons fait sous le simulateur Castalia/OMNeT++ et nous finiront par évaluer les performances des protocoles en termes d'efficacité de stockage et de fiabilité de lecture.

CHAPITRE IV

Evaluation de protocoles de stockage et de lecture d'informations dans la matière communicante

I. Introduction

La simulation des réseaux de capteurs sans fil est un outil important pour les chercheurs. Beaucoup d'universités mais aussi d'entreprises utilisent des simulateurs de réseaux différents car chacun répond à des besoins spécifiques. Parmi les simulateurs existants, nous avons choisi le simulateur Castalia/OMNeT++ pour tester le fonctionnement et les performances de nos protocoles. Ce simulateur nous permet d'éprouver nos protocoles dans des situations proches de la réalité. Dans ce chapitre, nous donnons les raisons qui nous ont encouragées à choisir ce simulateur. Ensuite, nous présentons le simulateur. Enfin, nous détaillons les résultats d'évaluation des protocoles USEE et RaWPG en tenant compte du contexte applicatif du préfabriqué en béton.

II. Les simulateurs de réseaux de capteurs sans fil

Un point commun entre tous les simulateurs est qu'ils permettent d'évaluer les performances d'un système avant son déploiement. Dans le cas de la simulation des réseaux de capteurs sans fil, les objectifs de base sont très diversifiés et concernent entre autres la validation des protocoles de routage ou encore des protocoles de sécurité. De nombreux outils de simulation sont utilisés dans les travaux scientifiques portant sur les réseaux de capteurs (Musznicki and Zwierzykowski, 2012) comme NS3, COOJA, Prowler, TOSSIM, SENSE et Castalia.

II.1. NS3

NS3 est un simulateur open-source à événements discrets, écrit en C++ et Python. Il est souvent utilisé dans un cadre académique pour des recherches portant sur les réseaux sans fil et filaires. Les nœuds dans le simulateur NS3 modélisent des piles protocolaires OSI complètes qui ne sont pas nécessaires pour représenter des nœuds de capteurs. En effet, un ensemble de protocoles de routage fixe, des protocoles de transport et des modèles d'application (comme des services web) sont fournis mais ils ne sont pas utilisés par les réseaux de capteurs. NS3 emploie un outil appelé *nam* pour la visualisation graphique des flux de communication obtenu par simulation. NS3 ne propose pas de modules de capture environnementale, et le format des paquets et les protocoles MAC sont différents de ceux qui sont utilisés par les plateformes de nœuds de capteurs. En outre, le modèle de consommation d'énergie développé est simpliste.

II.2. COOJA

COOJA est un simulateur inclus dans les outils de système d'exploitation Contiki et principalement développé pour simuler les nœuds de capteurs fonctionnant avec ce système. COOJA est disponible en tant que partie d'Instant Contiki, qui est une machine virtuelle contenant un environnement de développement pour Contiki. Ce simulateur permet l'émulation de différents capteurs sur lesquels seront chargés le système d'exploitation et les

applications COOJA. Avec ce simulateur, un ensemble de nœuds fonctionnant avec différents matériels et différentes applications peut coexister dans la même simulation.

II.3. Prowler

Prowler est un simulateur à événements (event-driven*)* de réseau de capteurs sans fil conçu pour fonctionner dans un environnement Matlab. Le simulateur, écrit à l'origine pour simuler les nœuds Berkeley MICA, est extensible aussi pour d'autres plateformes de capteurs. Prowler est utilisé principalement pour tester les protocoles de routage, le taux de réception et le délai d'acheminement des paquets. Prowler est mis en œuvre dans le langage Matlab (m-file). L'avantage de l'environnement Matlab est le prototypage facile des applications, l'intégration de différents algorithmes d'optimisation et l'interface graphique qui apporte une bonne capacité de visualisation. Le support des modèles de simulation comme la propagation radio est proposé sous forme de plugins m-file. Toute fois, Prowler ne modélise pas la consommation d'énergie. De plus, un seul protocole TinyOS MAC a été mis en œuvre par défaut.

II.4. TOSSIM

TOSSIM est un simulateur de système d'exploitation TinyOS. Il permet de simuler le comportement d'un nœud fonctionnant avec ce système. TOSSIM simule la plateforme de nœuds Berkeley MICA. Bien qu'il permette la simulation de milliers de nœuds dans un réseau de capteurs, il ne modélise ni un environnement réel ni la consommation d'énergie. TOSSIM emploie un outil nommé TinyViz, basé sur Java, qui fournit l'interface graphique pour visualiser, contrôler et interagir avec la simulation.

II.5. SENSE

SENSE a été développé par l'institut polytechnique Américain de Rensselaer en 2004. Dans SENSE, les éléments du réseau sont représentés dans une architecture de composants fournis sous une extension C ++ appelée CompC ++. Les modules disponibles incluent le modèle de batterie et des protocoles de couche MAC (NullMAC et IEEE 802.11). Les protocoles de couche réseau implémenté comprennent le Flooding et le protocole AODV. Ce simulateur utilise l'outil de visualisation iNSpect pour analyser la sortie des simulations et pour effectuer une animation des flux des messages, et l'outil G-Sense pour l'interfaçage graphique.

II.6. Castalia

Castalia est un simulateur de réseaux de capteurs sans fil, de réseaux BAN (Body Area Networks) et plus 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 du réseau 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 ICT Australia) en 2006. En 2007, il est devenu public en tant que projet open source sous la licence publique académique.

Après une étude des simulateurs existants, nous avons choisi comme simulateur Castalia sous la plateforme OMNeT++ car il offre un environnement de simulation proche de la réalité pour les réseaux de capteurs sans fil (Pediaditakis et al., 2010) (Rastegarnia et al., 2011). De plus, il est facilement paramétrable et peut implémenter une grande variété de plateformes de nœuds de capteurs.

III. Le simulateur Castalia

Notre implémentation des protocoles USEE et RaWPG a nécessité une étude approfondie des différents modules constituant le simulateur Castalia. L'étude du code source de certains protocoles (MultipathRings, AODV, LEACH) implémentés par différents chercheurs a été nécessaire pour comprendre les méthodes et les fonctions utiles pour notre travail.

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 messages générés par ce module.

Un nœud dans un réseau de capteurs sous Castalia est composé de différents modules comme l'indique la figure 57. Les flèches pleines représentent les messages échangés et les flèches en pointillés sont des appels à des fonctions du module.

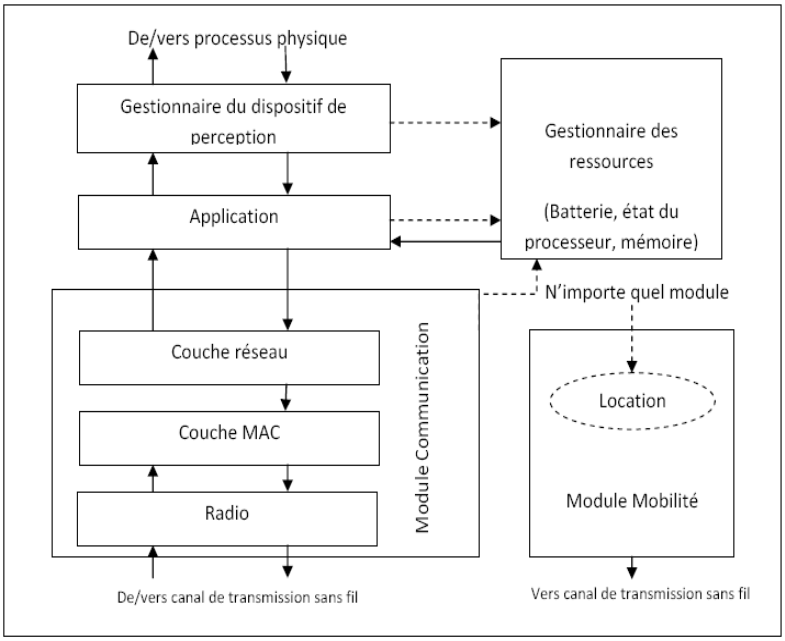


Figure 57. Structure d'un nœud sous Castalia (Boulis, 2011)

III.1. Le module canal sans fil

Ce module offre différents paramètres pour simuler un canal sans fil réel comme la propagation radio, l'interférence et le type de mobilité des nœuds.

III.2. Le module radio

Ce module supporte plusieurs états du nœud de capteurs : en transmission, en réception, en écoute, en veille. Il est possible de faire varier les puissances et les délais de transmission pour les différents états.

III.3. Le module MAC

Le module MAC (Medium Access Control) est une partie importante dans le comportement du nœud. La première motivation pour le développement de Castalia a été de tester le réglage du protocole MAC dans des conditions réelles du canal radio. Les protocoles MAC implémentés sous Castalia sont : TunableMAC, T-MAC, SMAC, IEEE 802.15.4 et Baseline-BAN-MAC.

III.4. Le module réseau

À partir de la version 1.2, le simulateur introduit le module réseau. Au début, les développeurs n'ont pas donné une importance à cet élément. Il n'y avait donc pas de module pour le routage, obligeant l'utilisateur à traiter le routage des paquets dans le module application. Castalia offre de base un seul protocole de routage « MultipathRings », mais il existe d'autres protocoles comme AODV et LEACH qui ont été implémentés par différents chercheurs.

III.5. Le module processus physique

Pour avoir un environnement de simulation proche de la réalité, le simulateur Castalia offre un module générique pour le processus physique de façon à fournir les données aux nœuds de capteurs. La base de ce modèle repose sur des valeurs des sources dont l'influence est diffusée dans l'espace. Les sources peuvent changer leur position et leurs valeurs.

III.6. Le module de gestion du dispositif de perception

C'est un module intermédiaire entre le module application et le module processus physique. L'application envoie les requêtes de lecture sur des valeurs ambiantes vers ce module, et ce dernier accède au processus physique pour lire la valeur. Ce module permet aussi de simuler la réception de données (évènements).

III.7. Le module de gestion des ressources

Ce module sauvegarde des traces des ressources des différents nœuds, en particulier l'énergie. Il soustrait l'énergie demandée par les différents modules et il effectue des statistiques sur l'utilisation de la mémoire.

III.8. Le module application

C'est le module dans lequel l'utilisateur effectue généralement beaucoup de changements pour implémenter de nouveaux algorithmes et des applications.

III.9. Le module mobilité

Le module de mobilité spécifie la manière dont les nœuds se déplacent 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.

Dans le reste de ce chapitre, nous évaluons et discutons les performances du protocole de dissémination USEE et du protocole de recherche d'information RaWPG proposés, dans le simulateur Castalia sous le système d'exploitation Linux Ubuntu. Nous allons tout d'abord commencer par étudier le cycle de vie du préfabriqué, et les informations qui y transitent. Ensuite, nous étudions l'organisation typique et idéale des micro-nœuds pour l'instrumentation d'un préfabriqué en béton. Enfin, les résultats de simulation sont présentés et discutés.

IV. Etude des informations du cycle de vie du préfabriqué à considérer

L'instrumentation du préfabriqué obéit classiquement à des exigences de mesures physiques (mesures de température, humidité, détection de fissures, etc.). De plus, dans notre contexte, la matière doit être capable de stocker des informations relatives à son cycle de vie. Il est donc nécessaire de définir le cycle de vie du préfabriqué, et d'identifier les informations nécessaires pour les différents acteurs intervenant tout au long de son cycle de vie. Ces informations peuvent être transmises par l'environnement ou alors produites par le préfabriqué lui-même, à l'aide des propriétés de capture d'informations, natives dans les réseaux de capteurs.

Dans cette étude, on considère le cycle de vie du préfabriqué qui est classiquement composé en trois grandes phases (figure 58) : *BOL (Beginning Of Life), MOL (Middle Of Life) et EOL (End Of Life).*

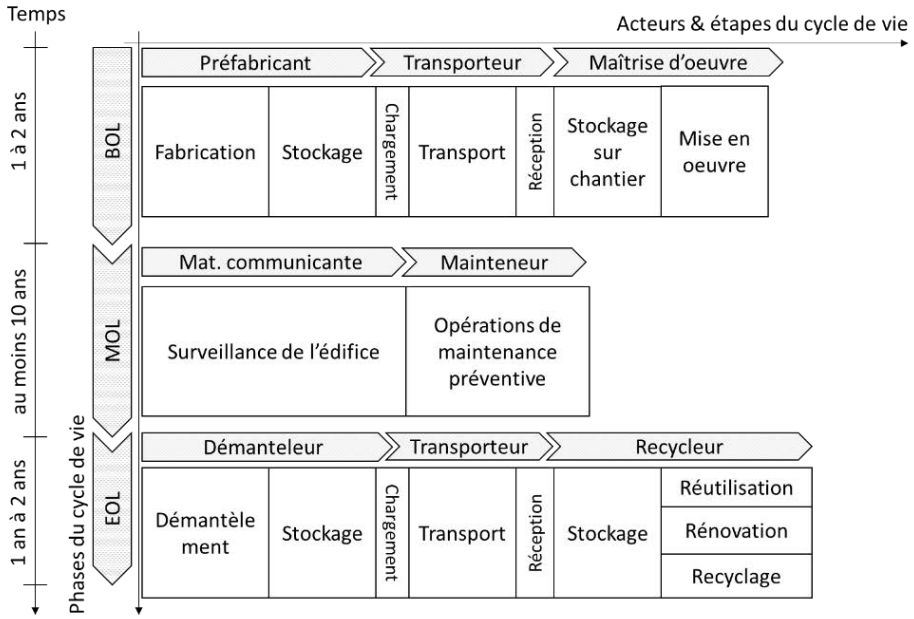


Figure 58. Cycle de vie du préfabriqué

IV.1. BOL (Beginning Of Life – début de vie)

Cette phase s'étale de la fabrication du préfabriqué jusqu'à sa mise en œuvre. Une étude précédente (Albin, 2014) sur la traçabilité des produits béton, menée avec l'aide d'un centre reconnu dans l'industrie du préfabriqué, a étudié finement cette phase, pour en dégager l'ensemble des informations présentées dans le tableau 5.

Tableau 5. Informations de la phase BOL

Phase	Etape	Informations	Champ base de données	Taille en octets	Nombre d'écritures	Nombre de lectures
BEGINNING OF LIFE (1 an à 2 ans)	Fabrication/stockage	Id Préfabriqué	INTEGER	2	1	
		Référence Préfabricant	VARCHAR(31)	31	1	
		Coordonnées Préfabricant	VARCHAR(31)	31	1	
		Longueur Préfabriqué	FLOAT	4	1	
		Largeur Préfabriqué	FLOAT	4	1	
		Hauteur Préfabriqué	FLOAT	4	1	
		Date de coulage	DATE	4	1	
		Résistance à la compression	FLOAT	4	1	
		Acier béton armé : Résistance ultime à la traction	FLOAT	4	1	
		Acier béton armé : Limite d'élasticité en traction	FLOAT	4	1	
		Acier précontraint : Résistance ultime à la traction	FLOAT	4	1	
		Limite conventionnelle d'élasticité	FLOAT	4	1	
		Tolérance en longueur	FLOAT	4	1	
		Tolérance en largeur	FLOAT	4	1	
		Tolérance en hauteur	FLOAT	4	1	
		Déviation angulaire d	FLOAT	4	1	
		Déviation angulaire Q	FLOAT	4	1	
		Défaut de planéité p	FLOAT	4	1	
		Flèche	FLOAT	4	1	
		Tolérance de flèche	FLOAT	4	1	
		référence fiche de déclaration environnementale du produit	VARCHAR(31)	31	1	
		adresse sur le web de la fiche de déclaration	VARCHAR(31)	31	1	
		adresse sur le web de la fiche de manutention du produit	VARCHAR(31)	31	1	
		plan de calepinage du produit	VARCHAR(31)	31	1	
		coordonnées livraison Produit	VARCHAR(127)	127	1	
		Température pour maturométrie*	INTEGER	2	336	14
	Chargement / Déchargement	Masse Préfabriqué				2
		Longueur Préfabriqué				2
		Largeur Préfabriqué				2
		Hauteur Préfabriqué				2
		fiche de manutention du produit				2
	Transport	coordonnées livraison Produit				2
		Réfrence transporteur	VARCHAR(31)	31	1	
		coordonnées transporteur	VARCHAR(127)	127	1	
		Date de livraison sur chantier	DATE	4	1	
	Stockage/Déstockage sur chantier	Masse Préfabriqué				2
		Longueur Préfabriqué				2
		Largeur Préfabriqué				2
		Hauteur Préfabriqué				2
		fiche de manutention du produit				2
		Identité sur chantier	INTEGER	2	1	
		Emplacement de calepinage	VARCHAR(31)	31	1	
	Mise en œuvre	plan de calepinage du produit				2
		Identité sur chantier				2
		Emplacement de calepinage				2
Légende			mémoire consommée (octets) :=	1250		
	informations écrites		taille moyenne donnée (octet) :=	18,7		
	informations lues		nombre d'accès en écriture minimal:=		341	
	informations produites par la matière		nombre d'accès en lecture minimal:=			42

(*) Les nombres de lecture et écriture sont obtenus pour une maturation de 14 jours (enregistrement de la mesure toutes les 1/2h (soit 336 fois), relevé une fois par jour par l'opérateur. On ne considère pas de traitement automatique de l'information).

Ces informations peuvent être liées aux propriétés physiques du préfabriqué (longueur, largeur, hauteur, différentes résistances mécaniques, etc.), à la logistique (coordonnées livraison produit et emplacement, date de livraison sur chantier, etc.), aux directives

environnementales (référence fiche de déclaration environnementale, adresse web de la fiche), à la manipulation du préfabriqué lorsqu'on le déplace (référence fiche de manipulation, adresse web fiche manipulation) ou à son montage sur site (plan de calepinage). À chaque étape de ce cycle de vie, les informations peuvent être écrites/modifiées (elles apparaissent alors en blanc dans le tableau), simplement lues (en gris) ou produites par la matière communicante elle-même, à l'aide des capteurs embarqués (en rayé blanc/gris). Par exemple, dans le tableau 5, on retrouve une information nommée « température pour maturométrie », qui possède une valeur produite par les capteurs eux-mêmes et stockée dans les nœuds. À chaque information correspond un type de données dont la taille en octet est indiquée.

Ce premier tableau permet d'identifier que la mémoire consommée par l'ensemble des données générées par cette phase est de **1250 octets**, la taille moyenne d'une donnée de **18,7** octets, le nombre d'accès en écriture de **341** et le nombre d'accès en lecture de **42**. De plus, des contraintes relevées par (Albin, 2014) précisent que le délai d'écriture requis doit être inférieur à **1 minute** et le délai de lecture inférieur à **5 secondes**, quelle que soit la phase considérée.

IV.2. MOL (Middle Of Life – usage)

Cette phase correspond à l'après-construction, lorsque le bâtiment est en usage. Dans ce cas, le bâtiment passe par deux étapes (surveillance de la structure & opérations de maintenance) qui se répètent pendant toute la phase (nous avons considéré qu'elle devait durer au minimum 10 ans, durée de la garantie décennale).

La surveillance de la structure est une phase active pour le réseau de capteurs qui va enregistrer à intervalles réguliers la présence de fissures, de température interne anormale et d'hygrométrie anormale du béton dans sa zone. Ces informations sont produites par chaque nœud.

La phase de maintenance va réutiliser des données inscrites dans la matière pour évaluer son état et procéder à des réparations si besoin. L'opérateur va ensuite rajouter les informations relatives à l'opération réalisée (date de l'opération, référence société, etc.).

Les nombres d'écriture et de lecture qui figurent dans le tableau 6 sont basés sur les deux hypothèses suivantes :

- *Surveillance de structure :* mesures en continu 1 fois/jour pendant 10 ans et relevé par un technicien 2 fois par an (cela fait donc 3650 écritures et 3650 lectures).
- *Opérations de maintenance :* maintenance préventive bisannuelle avec relevé des données et mise à jour si nécessaire (20 lectures/écritures).

Ce deuxième tableau permet d'identifier que la mémoire consommée par l'ensemble des données générées par cette phase est de **31950 octets**, la taille moyenne d'une donnée de **40,9** octets, le nombre d'accès en écriture de **11150** et le nombre d'accès en lecture de **11430**.

Tableau 6. Informations de la phase MOL

Phase	Etape	Informations	Champ base de données	Taille en octets	Nombre d'écritures	Nombre de lectures
MIDDLE OF LIFE (10 ans minimum)	Surveillance de la structure	Température interne anormale	INTEGER	2	3650	3650
		Détection de fissures	BOOLEAN	1	3650	3650
		Hygrométrie anormale	FLOAT	4	3650	3650
	Opérations de maintenance de la structure	Date de l'opération de maintenance	DATE	4	20	20
		Référence société de maintenance	VARCHAR(31)	31	20	20
		Coordonnées société de maintenance	VARCHAR(127)	127	20	20
		Désignation opérateur	VARCHAR(31)	31	20	20
		Opérations réalisées	VARCHAR(127)	127	20	20
		Id Préfabriqué	INTEGER			20
		Référence Préfabricant	VARCHAR(31)			20
		Longueur Préfabriqué	FLOAT		20	20
		Largeur Préfabriqué	FLOAT			20
		Hauteur Préfabriqué	FLOAT		20	20
		Masse du Préfabriqué	FLOAT		20	20
		Date de coulage	DATE			20
		Résistance à la compression	FLOAT		20	20
		Acier béton armé : Résistance ultime à la traction	FLOAT			20
		Acier béton armé : Limite d'élasticité en traction	FLOAT			20
		Acier précontraint : Résistance ultime à la traction	FLOAT			20
		Limite conventionnelle d'élasticité	FLOAT			20
		Tolérance en longueur	FLOAT			20
		Tolérance en largeur	FLOAT			20
		Tolérance en hauteur	FLOAT			20
		Déviation angulaire d	FLOAT			20
		Déviation angulaire Q	FLOAT			20
		Défaut de planéité p	FLOAT		20	20
		Flèche	FLOAT			20
Légende		mémoire consommée (octets) :=		31950		
	informations écrites	taille moyenne donnée (octet) :=		40,9		
	informations lues	nombre d'accès en écriture minimal:=			11150	
	informations produites par la matière	nombre d'accès en lecture minimal:=				11430

IV.3. EOL (End Of Life – fin de vie)

Cette phase correspond à la phase de recyclage du bâtiment, lorsqu'il s'apprête à être démantelé. Il est alors possible d'utiliser l'instrumentation de chaque préfabriqué pour faciliter le processus, à condition bien entendu que le réseau de capteurs soit toujours fonctionnel. Dans cette phase, le préfabriqué est alors démantelé, envoyé vers un recycleur qui décide du mode de retraitement le plus adéquat (réutilisation, rénovation ou recyclage complet). Les acteurs de cette phase ont besoin du maximum d'informations disponibles en particulier celles ayant trait aux propriétés initiales du préfabriqué, et celles issues des opérations de maintenance. Les données rajoutées au cours de cette phase sont essentiellement logistiques (tableau 8).

Ce troisième tableau permet d'identifier que la mémoire consommée par l'ensemble des données générées par cette phase est de **486 octets**, la taille moyenne d'une donnée de **34,7** octets, le nombre d'accès en écriture de **9** et le nombre d'accès en lecture de **81**.

IV.4. Synthèse

Au final, comme présenté dans le tableau 8, l'ensemble des données générées par le cycle de vie du préfabriqué représenté **33686 octets (~ 34 ko)** avec une taille moyenne de données de **29,02 octets (~30 octets)**. Le nombre minimal d'opérations d'écriture est de **11500**, alors que le nombre de lecture est de **11553**. Les délais de lecture/écriture sont ceux énoncés

précédemment. L'importance de l'information des différentes données n'est pas fixée et peut varier entre 0 et 1 en fonction du choix de l'utilisateur (comme expliqué dans le chapitre 1).

Ce tableau final contient l'ensemble des contraintes informationnelles imposées par notre contexte de travail. Les choix effectués dans les sections suivantes devront permettre de répondre à ces exigences.

Tableau 7. Informations de la phase EOL

Phase	Etape	Informations	Champ base de données	Taille en octets	Nombre d'écritures	Nombre de lectures
END OF LIFE (1 à 2 ans)	Démantèlement	Référence Préfabricant	VARCHAR(31)			1
		Masse Préfabriqué	FLOAT			1
		Longueur Préfabriqué	FLOAT			1
		Largeur Préfabriqué	FLOAT			1
		Hauteur Préfabriqué	FLOAT			1
		Date de coulage	DATE			1
		Résistance à la compression	FLOAT			1
		Acier béton armé : Résistance ultime à la traction	FLOAT			1
		Acier béton armé : Limite d'élasticité en traction	FLOAT			1
		Acier précontraint : Résistance ultime à la traction	FLOAT			1
		Limite conventionnelle d'élasticité	FLOAT			1
		Historique des opérations de maintenance	VARCHAR(x)			20
		référence fiche de déclaration environnementale du produit	VARCHAR(31)			1
		adresse sur le web fiche de déclaration environnementale	VARCHAR(127)			1
		adresse sur le web de la fiche de manutention du produit	VARCHAR(127)			1
		Référence démanteleur	VARCHAR(31)	31	1	
		coordonnées démanteleur	VARCHAR(127)	127	1	
		Date de démantèlement	DATE	4	1	1
	Chargement / Déchargement	Masse Préfabriqué	FLOAT			2
		Longueur Préfabriqué	FLOAT			2
		Largeur Préfabriqué	FLOAT			2
		Hauteur Préfabriqué	FLOAT			2
		fiche de manutention du produit	FLOAT			2
	Transport	coordonnées livraison Produit	FLOAT			2
		Réfrence transporteur	VARCHAR(31)	31	1	
		coordonnées transporteur	VARCHAR(127)	127	1	
		Date de livraison sur chantier	DATE	4	1	
	Recyclage	Toutes les Informations issues du démantèlement	-			34
		Référence recycleur	VARCHAR(31)	31	1	
		coordonnées recycleur	VARCHAR(127)	127	1	
		Date de recyclage	DATE	4	1	
Légende			mémoire consommée (octets) :=	486		
	informations écrites		taille moyenne donnée (octet) :=	34,7		
	informations lues		nombre d'accès en écriture minimal:=		9	
			nombre d'accès en lecture minimal:=			81

Tableau 8. Tableau récapitulatif de l'étude et des contraintes idoines sur l'instrumentation

Récapitulatif de l'étude sur le cycle de vie du produit préfabriqué	
Donnée	**Résultat**
Mémoire consommée totale (en octets):=	33686
Taille moyenne d'une donnée (en octets) :=	29,02
Nombre minimal d'opérations d'écriture :=	11500
Nombre minimal d'opérations de lecture :=	11553
Délai d'écriture requis :=	< 1 min
Délai de lecture requis :=	< 5s

V. Environnement et paramètres de simulation

Un bloc de matière de dimension 20mx20m est simulé dans cette étude. Tous les nœuds sont stationnaires et déployés uniformément dans la matière (sous forme d'une grille). Les caractéristiques du canal radio sans fil comme le bruit du signal et l'interférence sont choisies pour simuler le modèle de canal radio sans fil le plus réaliste de Castalia en se basant sur un

modèle d'interférence additive. Les paramètres de l'algorithme d'inondation counter-based *Cth* et T_{max} du protocole USEE sont fixés respectivement à 4 et 200 ms, ce qui autorise un compromis entre haute accessibilité et efficacité énergétique dans un réseau de capteurs dense comme (Izumi et al., 2008) et (Mekki et al., 2014b) le montrent. Les poids de la fonction de coût de liaison α, β et γ du protocole RaWPG sont fixés respectivement à 3, 2 et 3 suivant la recommandation de (Othman and Yahya, 2010) pour avoir la meilleure qualité de service dans la recherche des chemins. Une donnée de taille 30 octets est utilisé dans cette étude. Les simulations ont été répétées 60 fois, et la moyenne des résultats est calculée pour chaque étude. Les paramètres utilisés dans cette étude de simulation sont résumés dans le tableau 9.

Tableau 9. Paramètres de simulation

Version Linux Ubuntu	12.04 LTS	
Version OMNeT++ et Castalia	OMNeT++ 4.1 et Castalia 3.2	
Dimension de la matière	20mx20m	
Distribution des noeuds	Uniforme, Grille	
Le paramètre "Sigma channel"	4	Canal radio sans fil la plus réaliste dans Castalia
Le paramètre "Bidirectional Sigma"	1	
Modèle d'interférence	2 (modèle d'interférence additive)	
Protocole MAC	T-MAC	
Cth	4	
T_{max}	200 ms	
Poids α, β et γ	3,2,3	
Taille de donnée	30 octets	
Nombre de répétition	60	

VI. Instrumentation d'un réseau de capteurs dans un préfabriqué en béton

Dans cette partie, nous présentons l'organisation typique et idéale des micro-nœuds pour l'instrumentation d'un préfabriqué en béton par le concept de matière communicante. Pour cela, nous étudions le choix des nœuds et de la densité des nœuds optimale dans le préfabriqué, de manière à répondre aux exigences imposées par le contexte de travail.

VI.1. Choix des micro-nœuds de capteurs

Nos travaux doivent utiliser des micro-nœuds de capteurs dont les dimensions sont très réduites pour les intégrer facilement au sein d'un préfabriqué sans modifier ni ses propriétés, ni son aspect extérieur. Différents micro-nœuds de capteurs ont été développés dans l'industrie (voir tableau 10) comme *µPart* (Beigl et al., 2005), *ECO* (Park et al., 2005), *ZN1* (Yamashita et al., 2006), *SAND* (Ouwerkerk et al., 2006), *SensorCube* (Torfs et al., 2004) et *Tyndall* (Harte et al., 2008).

SAND, Tyndall et µPart sont les micro-nœuds ayant des dimensions les plus réduites (1 cm^3). Cependant, µPart est un nœud sans fonction de réception (un nœud de transmission

d'informations) et ne permet pas la communication multi-sauts dans un réseau de capteurs. Tyndall est un nœud à durée de vie très limité (10 heures) et nécessite un chargement en continue de sa batterie ce qui semble difficile à réaliser lorsque le nœud est intégré dans le béton. De plus, SAND est aussi un nœud à durée de vie très limité (432 heures). En analysant, les caractéristiques techniques des autres nœuds SensorCube, ZN1 et ECO, SensorCube présente quant à lui différents avantages, à savoir :

- Faible consommation d'énergie.
- Différents capteurs utiles pour notre contexte applicatif : température et humidité. Il n'a pas de capteurs de détection de fissures intégrés, mais des travaux comme (Grosse et al., 2006) proposent des solutions qui pourraient être intégrées à SensorCube.
- Taille grande de mémoire 120 Ko. Comme discuté précédemment (tableau 8), cela permet au nœud SensorCube de recevoir la totalité des informations (34 Ko) générées pendant le cycle de vie du préfabriqué.
- Il repose sur une architecture de composants électroniques extensibles. Les modules électroniques (microprocesseur, mémoire, capteurs, batterie, radio, etc.) sont superposés verticalement les uns sur les autres comme le montre la figure 59. On peut y ajouter des capteurs, mémoire et batteries avec comme inconvénient en contrepartie d'augmenter la taille du nœud.
- Il possède un module de gestion d'énergie qui supporte une source d'énergie extérieure tel qu'une pile, une batterie, ou un module de récupération d'énergie (Energy Scavenging module).

Au regard de tous ces avantages, le micro-nœud de capteurs SensorCube a été choisi et modélisé dans le simulateur Castalia. La puissance de transmission la plus faible de SensorCube c'est-à-dire -25 dbm a été choisie pour notre étude, afin de réduire la consommation d'énergie des transmissions et limiter le rayon de couverture radio de chaque nœud.

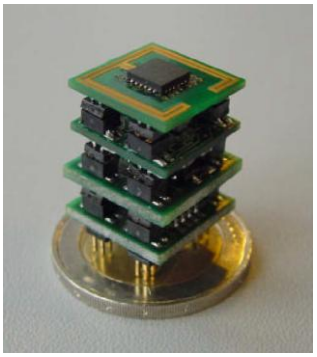


Figure 59. Structure hiérarchique du SensorCube mise sur une pièce 2 euro comme référence de taille (Torfs et al., 2004)

Tableau 10. Résumé des micro-nœuds de capteurs sans fil

	µPart	ECO	ZN1	SAND	SensorCube	Tyndall
Développeur	Karlsruhe Institute of Technology (Germany)	Center for Embedded Computer Systems (University of California- USA)	Hitachi Central Research Laboratory (Japan)	Philips Research (Netherlands)	IMEC and Holst Center Research (Netherlands)	Tyndall National Institute (Ireland)
Dimension	10x10x10 mm	12x12x7 mm	15x15x4 mm	10x10x10 mm	14x14x18 mm	10x10x10 mm
MCU	rfPIC16F675	Synopsys DW8051	Renesas H8S/2218	CoolFlux DSP	MSP430	8051 MCU
Fréquence	868 MHz	2.4 GHz	2.4 GHz	2.4G Hz	2.4 GHz	433/868/915 MHz
Bande passante	19.2 Kbps	1 Mbps	250 Kbps	250 Kbps	1 Mbps	50 Kbps
RAM	64 octets	4 Ko	12 Ko	10 Ko	4 Ko	4 Ko
Mémoire	1.4 Ko	32 Ko	128 Ko	64 Ko	120 Ko	8 Ko
Capteurs	Accéléromètre Température Luminosité	Accéléromètre	Température	Accéléromètre	Température Humidité Luminosité	Température Humidité Luminosité Accéléromètre
Mode active	340 µA	3 mA	6 mA	15.8 mA	2 mA	688 µA
Mode non active	0.8 µA	2 µA	1 µA	0.525 mA	2.5 µA	3.3 µA
Mode de transmission	11.2 mA	10.5 mA	24.9 mA	11 mA	13 mA	10.35 mA
Mode de réception	*Sans mode de réception*	19 mA	25.5 mA	19.7 mA	18 mA	13.32 mA

VI.2. Validation de la propagation du signal radio 2.4 GHZ dans la matière béton

Le nœud de capteurs SensorCube fonctionne dans la bande de fréquence 2.4 GHZ. Afin de tester la propagation de ce type du signal radio dans le béton, nous avons utilisé deux nœuds de capteurs MicaZ disponible dans notre laboratoire. Le nœud MicaZ fonctionne avec la même bande de fréquence 2.4 GHZ. Pour tester la communication radio dans le béton, nous avons configuré un nœud comme émetteur et l'autre comme récepteur. Puis, nous avons mis l'émetteur sur la face d'un mur en béton d'un bâtiment, et nous avons vérifié la réception de messages par le nœud récepteur mis sur l'autre face du mur comme l'illustre la figure 60. L'expérience a permis de valider la connectivité entre l'émetteur et le récepteur à travers le béton. Ainsi, le signal radio de type 2.4 GHZ peut se propager dans un préfabriqué en béton. Mais, des expériences plus précises doivent être faites pour une validation finale de ce fonctionnement par l'intégration des nœuds dans un préfabriqué réel (développement d'un prototype en béton).

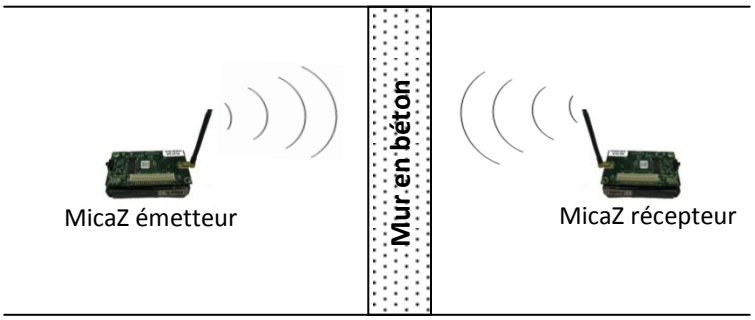


Figure 60. Validation de la propagation du signal radio 2.4 GHZ dans le béton

VI.3. Etude de la densité des nœuds de capteurs dans un préfabriqué

Nous étudions dans cette section la densité idéale des nœuds pour obtenir les meilleures performances d'une part de dissémination probabiliste uniforme de l'information et d'autre part de lecture de celle-ci.

VI.3.1. Etude de la densité des nœuds en fonction de l'uniformité de stockage probabiliste d'USEE

L'uniformité de dissémination dans le préfabriqué est un problème clé de nos travaux. On rappelle que quelle que soit la densité des nœuds, ils sont déployés dans le préfabriqué d'une manière uniforme (nœuds placés dans le préfabriqué avant le coulage du béton). Afin de déterminer la densité idéale des nœuds dans le préfabriqué, nous supposons donc que si l'information existe dans le voisinage de chaque nœud du réseau de capteurs, alors forcément l'information existera dans tout le préfabriqué (dissémination uniforme). Dans ce qui suit, nous étudions théoriquement et par simulation la densité des nœuds qui assure l'existence des informations dans tous les voisinages.

a) Etude théorique

Soit un réseau de capteurs à N nœuds dans le préfabriqué. Suite à la dissémination d'une information par le protocole USEE, ce réseau contient deux types de nœuds : les nœuds qui ont stocké l'information et les nœuds qui n'ont pas stocké l'information. Soit :

- e_1: nœud contenant l'information.
- e_2: nœud vide d'information.
- E_1 : {ensemble des nœuds du type e_1 qui stockent l'information}
- E_2 : {ensemble des nœuds du type e_2 qui ne stockent pas l'information}

La probabilité qu'un nœud stocke l'information est égale à P, d'où :

- $R = Card(E_1) = N \times P$: nombre total des nœuds du type e_1 qui stockent l'information.
- $N\text{-}R = Card(E_2) = N \times (1\text{-}P)$: nombre total des nœuds du type e_2 qui ne stockent pas l'information.

Soit n le nombre de nœuds dans le voisinage d'un nœud de capteurs dans le préfabriqué. Les n nœuds se composent évidement de nœuds e_1 et e_2. En statistique, le voisinage de taille n représente un tirage de n nœuds parmi le nombre de nœuds total N. Attachons maintenant à e_1

et e_2 un nombre X prenant la valeur 1 si e_1 est tiré et la valeur 0 si c'est e_2 qui est tiré. X est une variable aléatoire discrète qui représente une variable de Bernoulli, dont la loi de probabilité est donnée par les deux probabilités : prob(X=1)=*P* et prob(X=0)=(1-*P*) dont la somme est égale à 1.

On s'intéresse maintenant à la variable aléatoire *k* définie comme « le nombre de nœuds e_1 parmi les *n* nœuds tirés ». Ces *n* nœuds correspondent au voisinage d'un nœud de capteurs donné, et p(*k*) correspond à la probabilité de trouver *k* fois l'information désirée dans le voisinage du nœud.

Le tirage des nœuds du voisinage représente dans notre cas un *tirage simultané (non ordonné et sans remise)* de *n* nœuds parmi N possibles. La loi de probabilité associée à ce tirage est donc une loi de *probabilité hypergéométrique* de paramètres associés *n*, P et N pouvant être définie par la relation suivante :

$$p(k) = \frac{C_R^k \times C_{N-R}^{n-k}}{C_N^n} = \frac{C_{nP}^k \times C_{n(1-p)}^{n-k}}{C_N^n}$$

On s'intéresse à calculer la probabilité p(*k*≥1) de trouver l'information dans le voisinage au moins une fois. Cela correspond à la probabilité de trouver un nœud e_1 au moins une fois parmi les *n* nœuds. La probabilité p(*k*) suit une loi hypergéométrique telle que :

$$\sum_{i=0}^{n} p(k = x_i) = 1\,, avec\ x_i = \{0,1,2,3,\dots,n\}$$

Cela donne :

$$\mathrm{p(k=0)} + \underbrace{\mathrm{p(k=1)} + \mathrm{p(k=2)} + \mathrm{p(k=3)} + \mathrm{p(k=4)} + \cdots + \mathrm{p(k=n)}} = 1$$

$$\mathrm{p(k=0)} + \overbrace{\mathrm{p(k \geq 1)}} = 1$$

D'où la probabilité p(*k*≥1) est :

$$\mathrm{p(k \geq 1) = 1 - p(k = 0)}$$

Finalement, on obtient:

$$\mathrm{p(k \geq 1)} = 1 - \frac{C_{N-R}^n}{C_N^n}$$

La figure 61 illustre la variation de p(k≥1) en fonction de la taille du voisinage *n* pour chaque probabilité de stockage *P*. Elle illustre bien que plus on augmente le nombre des nœuds *n* dans chaque voisinage, plus la probabilité de trouver l'information dans chaque voisinage augmente. Ce calcul démontre qu'au delà d'un voisinage de 40 nœuds, l'information existe dans chaque voisinage quelque soit la probabilité de stockage *P* (voir figure 61). En fait, le nombre de nœuds dans le voisinage est déterminé en fonction de la puissance radio de chaque nœud et du nombre total de nœuds déployés régulièrement dans le préfabriqué.

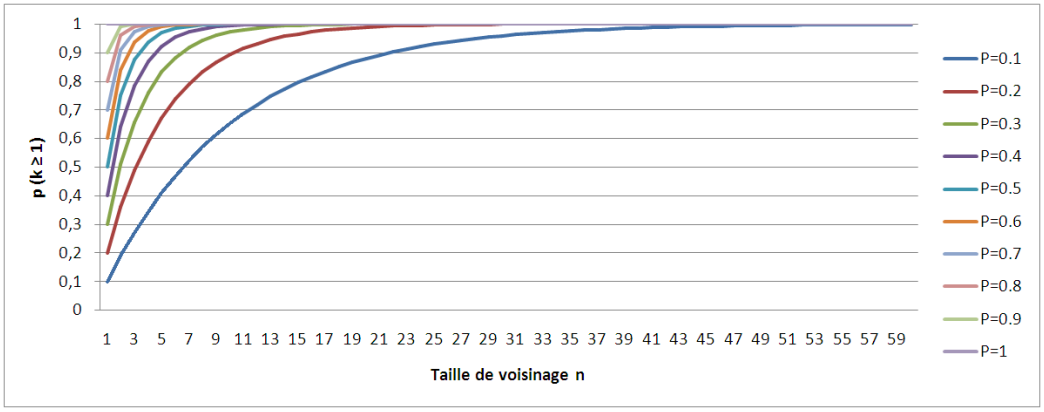


Figure 61. Variation théorique de probabilité d'existence d'information se trouvant au moins une fois dans le voisinage

b) Etude par simulation

L'objectif de cette étude est de valider les résultats obtenus théoriquement dans le paragraphe précédent. Pour cela, nous disséminons une information à l'aide du protocole USEE dans un réseau dont nous faisons varier la taille du voisinage *n*. Comme dit précédemment, la puissance de transmission de chaque nœud est fixé à -25 dbm. La taille du voisinage *n* est fonction du nombre total de nœuds dans le réseau.

La figure 62 illustre l'influence de la variation de *n* sur la probabilité d'existence de l'information dans le voisinage d'un nœud. La figure montre en effet que, lorsqu'on augmente le nombre de nœuds *n* du voisinage, le nombre de nœuds contenant l'information dans leur voisinage augmente : cela donne une probabilité de plus en plus grande de trouver l'information dans chaque endroit du préfabriqué. En suivant cette courbe, toutes les données sont présentes dans le voisinage de la quasi-totalité des nœuds de la matière (plus que 80% et 90% des voisinages contiennent l'information) pour des probabilités de stockage supérieure à 0.2 et pour les densités de voisinage *n*=25, *n*=37 et *n*=45. Par contre, la probabilité d'existence de l'information est faible pour des densités de voisinage très réduites dans le préfabriqué comme *n*=9 (entre 31% et 62%) et *n*=15 (entre 47% et 85%). En conclusion, un grand nombre de nœuds dans chaque voisinage dans le réseau de capteurs augmente la probabilité de trouver l'information dans toutes les parties du préfabriqué. Ainsi, cela permet d'assurer l'existence des informations dans la plupart des voisinages de nœuds pour toutes les probabilités de stockage, avec comme incidence de faciliter et d'accélérer l'exécution du processus de recherche et de lecture d'informations.

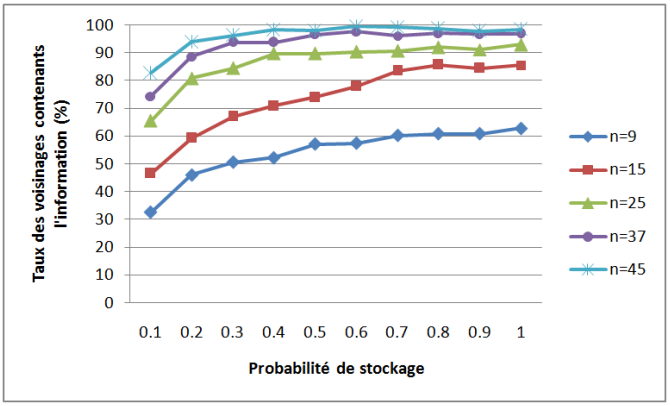


Figure 62. Taux d'existence de l'information dans les voisinages pour différentes tailles de voisinage *n*

VI.3.2. Etude de la densité des nœuds en fonction de l'efficacité de la sélection des nœuds de bordure dans RaWPG

L'objectif de cette section est d'étudier l'influence de la variation de la taille de voisinage sur l'efficacité de l'algorithme de sélection des nœuds de bordure dans RaWPG. La validation de l'efficacité de cet algorithme est une phase primordiale pour l'exécution du protocole de recherche RaWPG.

Les résultats que nous souhaitons obtenir sont de type graphique pour avoir une appréciation visuelle sur la sélection des nœuds de bordure dans le voisinage. La figure 63 donne le résultat de l'algorithme de sélection des voisins de bordure : les voisins de bordure sélectionnés par un nœud X (en noir) apparaissent en rouge.

La figure 63 illustre le résultat de simulation pour les densités de voisinage n = 9, 15, 25, 37 et 45. La figure montre que la détection des nœuds de bordure est plus précise pour une grande densité de nœuds dans le voisinage, les densités n=25, n=37 et n=45 offrent une meilleure sélection que celui pour n=9 et n=15. Cela peut se comprendre par le fait que les voisinages denses permettent de mieux distinguer les nœuds qui ont des faibles cardinalités d'intersection des voisinages, et qui représentent donc des voisins distants.

En conclusion, l'algorithme fonctionne parfaitement uniquement pour une grande densité dans le voisinage d'un nœud au sein du préfabriqué. Une faible densité dans le voisinage ne satisfait pas les résultats attendus par l'algorithme. En conclusion, le facteur de densité de voisinage prédomine dans la précision de l'algorithme.

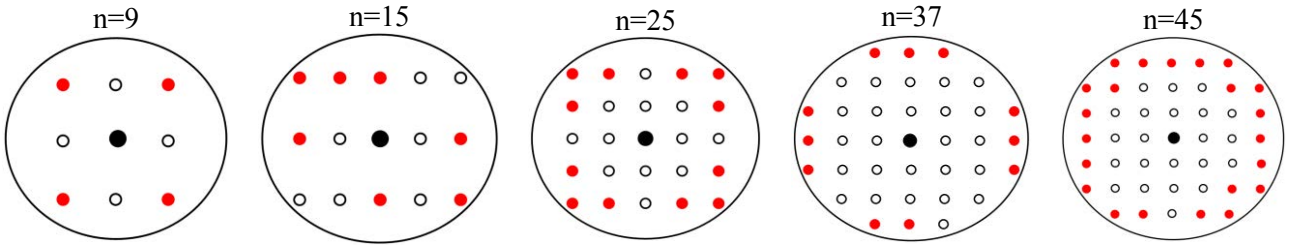


Figure 63. Résultats de sélection des nœuds voisins distants (bordures)

Dans le reste de ce chapitre, nous présentons dans un premier temps les résultats de performances du protocole de dissémination USEE puis dans un second temps ceux du protocole de lecture RaWPG. Une synthèse est donnée en guise de conclusion à la fin du chapitre.

VII. Evaluation de performances du protocole de dissémination USEE

Cette section détaille les résultats de dissémination et de stockage d'informations en utilisant le protocole de dissémination USEE présenté au chapitre précédent.

VII.1. Plan d'évaluation des performances

Pour évaluer les performances du protocole USEE, nous utilisons le nombre de nœuds dans chaque voisinage *n*=25. Un ensemble de 2500 nœuds est distribué uniformément dans le préfabriqué simulé. Les nœuds sont présents dans la matière au nombre de 25 nœuds tous les 4 m^2. L'évaluation des performances du protocole est faite sous forme de deux études :

- *Etude de la capacité de stockage de la matière* : cette étude permet d'analyser l'avantage du mécanisme de réplication controlé *NS* dans notre processus de dissémination USEE.
- *Comparaison d'USEE avec des protocoles de la littérature* : pour cela, nous avons choisi les protocoles DEEP (Vecchio et al., 2010), Supple (Viana et al., 2010) et RaWMS (Bar-Yossef et al. 2008) présentés dans le chapitre 2 car ils proposent aussi la réplication d'informations uniformément dans l'ensemble du réseau. Cela nous permet de bien montrer l'intérêt de nos propositions pour le contexte particulier du préfabriqué par rapport à des protocoles existants.

VII.2. Critères d'évaluation des performances

Cinq mesures de performances principales sont utilisées pour évaluer le protocole USEE. Ces mesures sont étudiées en fonction de la probabilité de stockage des informations :

- *L'uniformité géographique de réplication* : c'est l'existence des informations disséminées à chaque endroit de la matière communicante (préfabriqué béton dans notre application). Comme montré précedemment en figure 62, pour une taille de voisinage donnée, notre approche assure une même probabilité de présence dans le voisinage d'un nœud pour toute information ayant la même importance. Dans notre cas d'étude, les nœuds suivent une répartition géographique fixe, sous forme de grille (ils sont en effet positionnés avant que le préfabriqué soit coulé). Cette mise en place uniforme, combinée à une maîtrise de la probabilité de présence, devrait conduire à une présence géographique uniforme de l'information dans la matière. Afin d'observer ce phénomène, la matière communicante sera partionnée en cellules rectangulaires distinctes (correspondant à une partie de la matière, comme si celle-ci avait été découpée en morceaux) et l'existence de l'information sera étudiée sur chacune de ces cellules. Nous considérons alors que plus le nombre de cellules contiennent l'information, plus la répartition géographique est

uniforme. L'uniformité est quantifiée par le taux d'existence de l'information, qui se calcule selon l'équation suivante :

$$taux\ d'existence = \frac{nombre\ de\ cellules\ ayant\ l'\ information}{nombre\ total\ de\ cellules}$$

- *Taux de réplication* : c'est le nombre total de nœuds qui stockent une information. La réduction de ce nombre dans chaque processus de dissémination, augmente la capacité de stockage de la matière. Le taux de réplication se calcule selon l'équation suivante :

$$taux\ de\ réplication = \frac{nombre\ de\ noeuds\ stockant\ l'information}{nombre\ total\ de\ noeuds}$$

- *Délai de dissémination* : c'est le temps écoulé entre l'instant de début de diffusion de l'information par le nœud maitre, et l'instant de réception de l'information par les derniers nœuds dans la matière.
- *Nombre de communications* : c'est le nombre de transmissions d'un message dans le réseau pendant le processus de dissémination. Cette mesure est un indicateur fort de la consommation d'énergie dans le réseau de capteurs. En effet, la communication est l'activité principale qui est responsable de la consommation de l'énergie d'un réseau de capteurs sans fil comme le montre la figure 64 (Anastasi et al., 2009) (Nechibvute et al., 2012).
- *Consommation d'énergie* : c'est l'énergie moyenne consommée par chaque nœud de capteurs dans la matière après plusieurs éxecutions du protocole USEE.

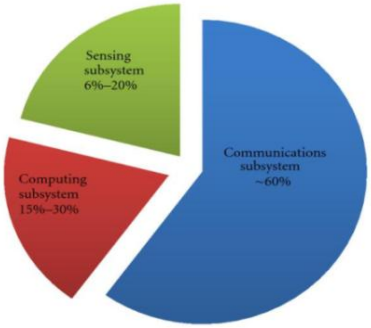


Figure 64. Consommation d'énergie des modules d'un nœud de capteurs (Nechibvute et al., 2012)

VII.3. Capacité de stockage de la matière

Dans cette section, nous avons supprimé le mécanisme de contrôle de réplication NS du protocole USEE (lignes 10, 11 et 12 de l'*Algorithme1* du protocole USEE) et nous avons comparé les résultats avec l'algorithme initial d'USEE en termes de taux de réplication (figure 66) et capacité de stockage de la mémoire de la matière (figure 67). La figure 65 donne les résultats de simulation de réplication d'une information dans le réseau pour différentes probabilité de stockage (différents niveaux d'importance *I*). En fait, suite à la simulation du protocole USEE, les nœuds stockant l'information ont été identifiés. Une grille graphique a été développée en C++ (API Qt) présentant ainsi le déploiement des nœuds dans

le préfabriqué. Chaque cellule de la grille représente un nœud dans le préfabriqué et chaque point noir représente un nœud qui a stocké l'information.

P=I	USEE sans NS	USEE avec NS
1		
0.8		
0.6		
0.4		
0.2		

Figure 65. Résultats de réplication d'USEE avec NS vs. USEE sans NS

Comme le montre la figure 66, le protocole USEE avec NS réduit le nombre de nœuds de stockage dans le réseau. Le taux de réplication est d'environ 2/3 par rapport au protocole sans NS. L'intérêt est que cela augmente la capacité du réseau à stocker des informations supplémentaires. La figure 67 montre que USEE avec NS pourrait atteindre 27000 opérations de stockage dans la matière, alors que, USEE sans NS ne permet que 7000 opérations de stockage. Le paramètre NS pourra donc garantir des capacités de stockage dans la matière

plus élevées. Cela montre l'avantage d'utiliser le champ d'entête NS dans USEE, car il permet de contrôler le taux de réplication des informations dans la matière et la résilience à la saturation de la mémoire des nœuds de capteurs. Avec NS, il n'y a pas d'information qui occupe toute la mémoire disponible des nœuds empêchant ainsi la réplication d'autres informations. Finalement, dans notre contexte (tableau 8), le nombre d'opérations de stockage doit être supérieur à 11500, ce qui élimine l'utilisation du protocole USEE sans NS. On peut aussi constater que le protocole USEE avec NS permet de doubler ce nombre d'opérations de stockage. Cela permet d'envisager d'enrichir de façon significative le système d'information associé au préfabriqué béton.

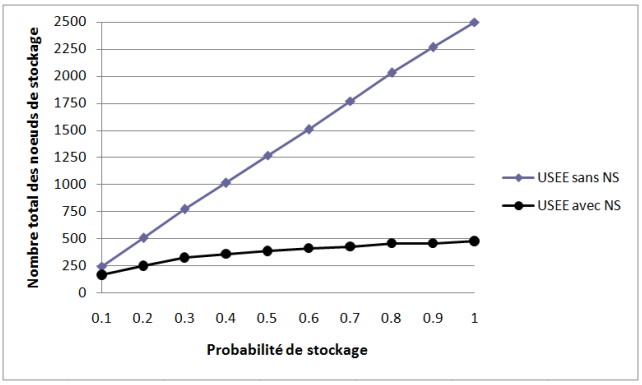


Figure 66. Taux de réplication d'USEE avec NS vs. USEE sans NS

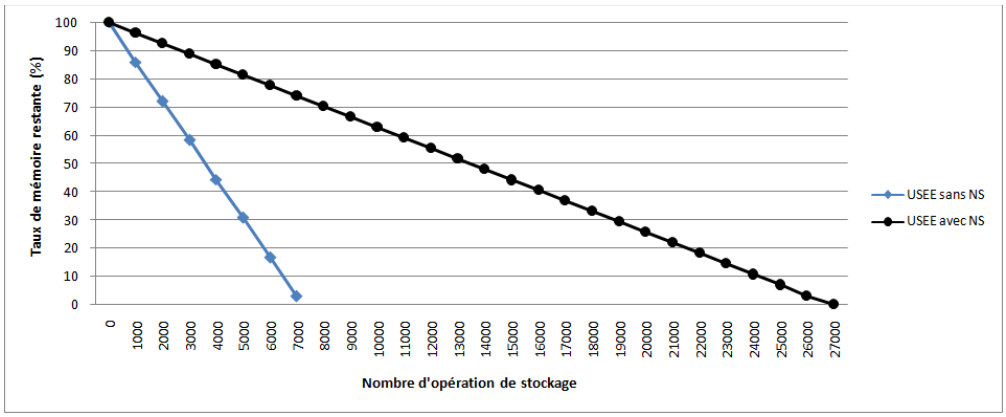


Figure 67. Capacité de stockage de la matière d'USEE avec NS vs. USEE sans NS

VII.4. USEE vs. Protocoles de littérature

Cette section compare les performances du protocole USEE avec les protocoles DEEP, Supple et RaWMS. Ces protocoles ont été choisis en raison de leurs similitudes avec USEE car ils proposent une réplication de données complète dans l'ensemble du réseau. Ces protocoles de la littérature fonctionnent également sur des réseaux à haute densité. Cependant, des adaptations ont été nécessaires pour que ces protocoles prennent en considération le niveau d'importance *I*. Les probabilités de stockage *W(i)* et *ps* (cf. chapitre 2) sont respectivement égales à *I* pour les protocoles Supple et DEEP. Pour le protocole Supple, la valeur prise pour *r* (cf. chapitre 2) est égale à $r = \sqrt{2500} = 50$. Pour le protocole RaWMS, le

nœud maître lance une marche aléatoire *RW* (cf. chapitre 2) *r* fois selon l'importance *I*. Différentes fonctions peuvent être utilisées pour déterminer le nombre de *RW* en utilisant *I* en tant que paramètre. Une fonction simple est adoptée et elle considère une valeur *r* pour chaque sous-ensemble d'intervalles de *I* comme cela est indiqué dans l'équation suivante (par exemple r=h(I) = 850 si $0{,}85<I\leq 0.9$) :

$$\forall\, I \in [0,1],\ h(I) = \begin{cases} 900 & \text{if} \quad 0.9<I\leq 1 \\ 850 & \text{if} \quad 0.85<I\leq 0.9 \\ 800 & \text{if} \quad 0.8<I\leq 0.85 \\ & \vdots \\ 50 & \text{if} \quad 0\leq I\leq 0.1 \end{cases}$$

Cette variation n'a pas d'effet négatif sur ces protocoles de la littérature évalués. Au contraire, elle conduit à une comparaison juste avec USEE pour l'analyse de l'uniformité, du taux de réplication, du délai de dissémination, du nombre de communications et de consommation d'énergie.

VII.4.1. Uniformité de réplication

Pour étudier l'uniformité de réplication, le préfabriqué simulé est logiquement divisé en 100 cellules (voir figure 68). L'étude consiste à évaluer le nombre de cellules logiques du réseau qui contient l'information. Cette expérience nous donne une idée de la façon dont les protocoles de dissémination distribuent les informations sur l'ensemble de la matière.

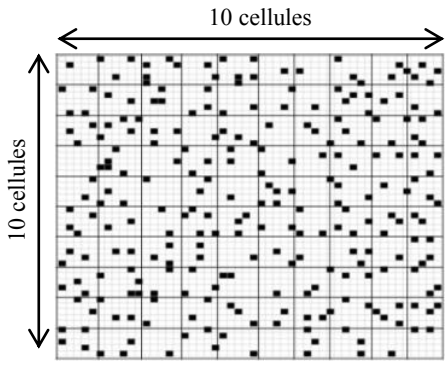


Figure 68. Division de la matière en 100 cellules de dimension

Les résultats de cette expérience sont présentés dans la figure 69. Elle présente le taux d'existence atteint par les protocoles évalués. Cette figure illustre que les protocoles USEE et DEEP disséminent les informations de façon plus uniforme que les autres. En effet, les protocoles USEE et DEEP répliquent les informations dans 91% des cellules pour I=0.1 et I=0.2, et 100% à partir de I=0.3. Pour le protocole Supple, la réplication atteint 100% des cellules uniquement à partir de I=0.4, et l'uniformité diminue pour des niveaux d'importance plus bas (0.3, 0.2 et 0.1) amenant des cellules à ne contenir aucune information.

En ce qui concerne le protocole RaWMS, les informations sont répliquées en fonction du nombre de *RW*. Lorsque l'on utilise un nombre élevé pour *RW*, les informations sont alors

stockées dans un grand nombre de cellules. On remarque qu'à partir de *I*=0.7, les informations se retrouvent dans 90% des cellules. Par contre, pour des *I* en dessous de 0.3, les informations sont contenues dans moins de 60% des cellules. Par conséquent, RaWMS a la plus faible performance en terme d'uniformité par rapport à USEE, DEEP et Supple. Cela montre que la stratégie de marche aléatoire n'est pas bonne dans notre contexte applicatif car elle génère de nombreuses zones vides d'information dans la matière communicante ce qui pourrait entraîner des pertes d'information au cours du cycle de vie de la matière. Par opposition, la stratégie de stockage probabiliste assure une bonne uniformité de l'information dans la matière.

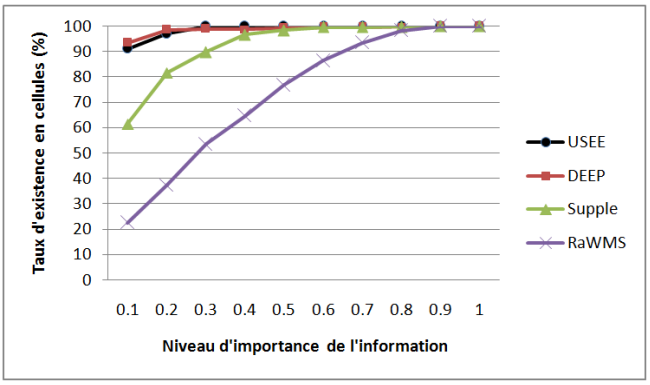


Figure 69. Uniformité de la réplication d'USEE vs. Protocoles de littérature

VII.4.2. Taux de réplication

Le taux de réplication de données dans tout le réseau est un indicateur de surcharge de la mémoire des nœuds du réseau. Aussi, le meilleur protocole doit montrer une distribution uniforme avec la plus faible surcharge de mémoire dans la matière. La figure 70 montre que le protocole USEE entraine la surcharge de mémoire la plus faible et qu'il maintient un taux de stockage contrôlé dans la matière pour toutes les valeurs d'importance (grâce au mécanisme de réplication contrôlé NS comme cela a été présenté dans la section VII.3 de ce chapitre). En effet, le nombre de nœuds stockant l'information avec un *I*=0.1 est de 168, 387 pour *I*=0.5 et 479 pour *I*=1. Pour les autres protocoles, il est clair que le taux de réplication a un comportement linéaire ce qui conduit à saturer la mémoire des nœuds, après plusieurs processus de stockage, et en particulier quand un niveau d'importance élevé est utilisé. Les protocoles DEEP et Supple sont les moins performants. À titre d'exemple pour *I* = 0.8, les informations sont stockées dans 1600 à 2150 nœuds pour Supple et 1750 à 2500 (c-à-d. tous les nœuds de la matière) pour DEEP. Cela conduit à la saturation rapide de la mémoire lorsque l'on réalise plusieurs phases de stockage dans la matière, c'est-à-dire que les nœuds ne peuvent plus stocker d'informations supplémentaires après plusieurs opérations d'écriture. Une solution pour pallier ce problème pourrait être de s'appuyer sur des techniques de compression de données, comme celles décrites dans (Marcelloni and Vecchio, 2008) et (Sadler and Martonosi, 2006). Les algorithmes de compression sont adaptés pour le stockage

réduit et les ressources limitées de nœuds de capteurs. Par exemple, dans (Marcelloni and Vecchio, 2008), le taux de compression atteint jusqu'à 70% pour un ensemble de données environnementales.

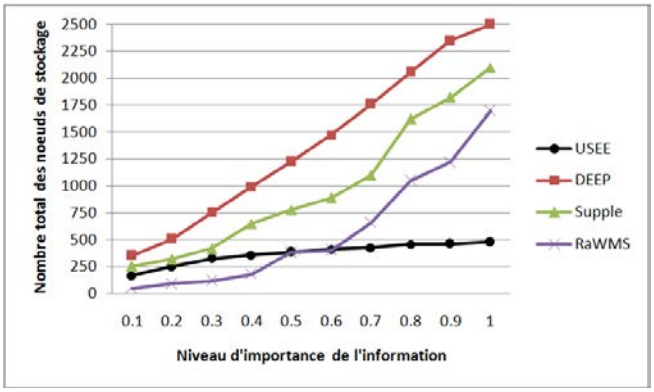


Figure 70. Taux de réplication d'USEE vs. Protocoles de littérature

VII.4.3. Délai de dissémination

Le délai de dissémination des données mesure le temps écoulé depuis le début du processus par le nœud maître jusqu'à la fin du processus de dissémination dans toute la matière communicante. Ce paramètre est influencé principalement par la stratégie de transmission du message à partir du nœud maître vers le reste du réseau. Comme le montre la figure 71, le protocole DEEP a le plus faible délai (min≈0.09s max≈0.13s) car il utilise l'inondation probabiliste, c'est-à-dire que chaque nœud diffuse directement le message avec une probabilité *pr*. Le protocole USEE utilise le délai d'attente *RAD* dans chaque nœud, ce qui provoque une augmentation du temps écoulé pour terminer le processus de dissémination (min≈0.28s et max≈0.39s). Malgré cela, le protocole USEE a un faible délai par rapport à Supple (min≈0.5s max≈0.6s) et RaWMS (min≈0.61s max≈5.4s). En utilisant les protocoles Supple et RaWMS, le nœud maître envoie plusieurs fois le message vers différents chemins. Après chaque transmission, le nœud maître ajoute un faible délai d'attente avant de commencer la prochaine transmission, provoquant ainsi un retard plus conséquent pour disséminer l'information.

Il est clair que le délai est indépendant du niveau d'importance pour les protocoles USEE, DEEP et Supple, puisque la stratégie de transmission est toujours la même, seule la probabilité de stockage dans chaque nœud varie selon le niveau d'importance. En utilisant les protocoles USEE et DEEP, le réseau exécute une seule fois l'inondation reposant respectivement sur counter-based et probabilistic-based pour tout niveau d'importance. Le protocole Supple a un nombre fixe de transmission (50 dans cette étude) pour tout niveau d'importance. Par contre, le délai augmente selon l'importance pour le protocole RaWMS car celui-ci effectue un nombre différent de transmission de marche aléatoire pour chaque niveau d'importance, ce qui a pour effet d'augmenter le temps de dissémination.

Si nous étendons maintenant la discussion en regard de notre contexte applicatif (tableau 8), la contrainte de délai sur l'écriture des informations dans le préfabriqué béton doit être inférieure à une minute. Cela montre que par rapport à nos résultats, c'est une contrainte faible puisque tous les protocoles sont en dessous de 6 secondes. Le fait que le protocole USEE ne soit pas celui qui obtient le meilleur résultat n'est donc pas un problème si nous nous référons à nos exigences applicatives.

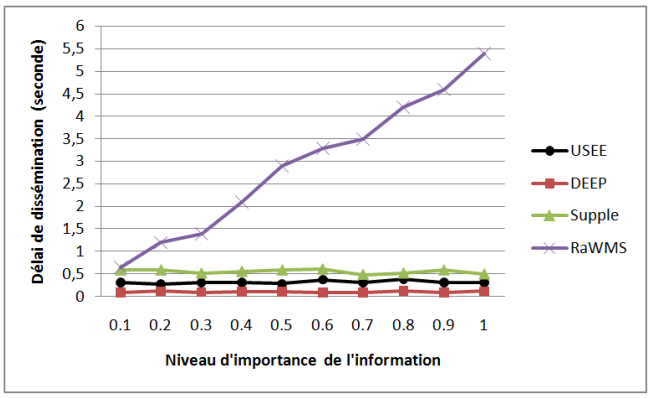


Figure 71. Délai de dissémination d'USEE vs. Protocoles de littérature

VII.4.4. Nombre de communications

La figure 72 donne le nombre total de messages transmis sur le réseau pour tous les protocoles évalués selon le niveau d'importance de l'information. Tout d'abord, on remarque que la charge de communication pour les protocoles USEE, DEEP et Supple sont indépendants du niveau d'importance. Les mécanismes de transmission de ces protocoles fonctionnent pour assurer la réception du message par tous les nœuds de réseau : inondation counter-based pour USEE, inondation probabiliste pour DEEP et transmission hiérarchique pour Supple. L'importance de l'information n'a donc aucun impact sur les mécanismes de transmission de ces trois protocoles. Elle influe uniquement sur la probabilité de stockage dans chaque nœud. Toutefois, la charge de communication augmente avec l'importance de l'information pour RaWMS. Dans ce cas, le nœud maître augmente le nombre r de transmission de messages pour chaque niveau d'importance, ce qui provoque une augmentation du nombre de fois où une marche aléatoire va être exécutée dans le réseau, ce qui entraîne une surcharge de communication durant la phase de dissémination.

Numériquement, on remarque que le protocole USEE transmet environ 25% de moins de messages que le protocole DEEP mais 30% de plus que le protocole Supple. Quand l'importance de l'information est inférieure à 0.2, RaWMS a le taux le plus bas car il utilise un faible nombre de r mais cela conduit également à une faible uniformité comme indiqué dans les sections précédentes. Cependant, le protocole RaWMS génère deux fois plus de messages que tous les autres protocoles à partir de I=0.7.

Le protocole USEE est plus performant que le protocole DEEP même s'ils utilisent tous les deux des mécanismes à inondation. Cet avantage s'expliquant par l'usage de la technique

couter-based dans USEE qui limite le nombre de rediffusions inhérentes à la technique probabilistic-based utilisée par le protocole DEEP.

De plus, le tableau 11 présente le nombre de communication du nœud maitre pour chaque protocole évalué. Comme décrit précédemment pour les protocoles Supple et RaWMS, le nœud maître transmet le message plusieurs fois vers différents chemins pour atteindre tous les nœuds du réseau. Le protocole Supple transmet le message 50 fois à travers des chemins dans la structure hiérarchique. De même, RaWMS transmet le message au minimum 50 fois (message de faible niveau importance) et au maximum 900 fois (message d'importance élevée). Par contre en utilisant les protocoles USEE et DEEP, le nœud maître envoie le message une seule fois à ses voisins. Le message est ensuite inondé dans tout le réseau. Evidemment, cela permet de disséminer les informations avec le minimum de coût énergétique pour le nœud maitre.

Comme la communication est le facteur impactant le plus sur la consommation d'énergie dans les réseaux de capteurs (Anastasi et al., 2009), tous ces résultats vont avoir une forte corrélation avec les performances du protocole USEE sur la durée de vie de la matière communicante.

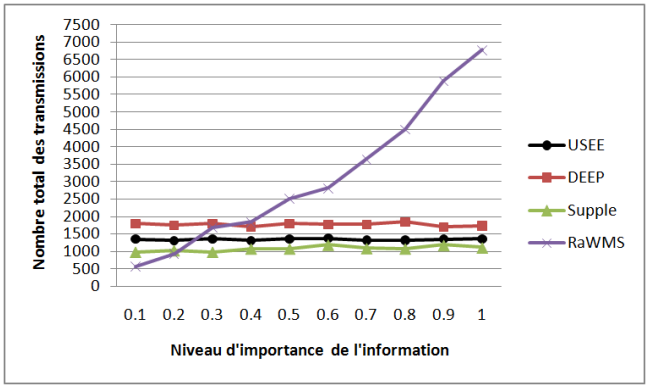


Figure 72. Nombre de communications d'USEE vs. Protocoles de littérature

Tableau 11. Nombre de transmissions du nœud maitre

	USEE	DEEP	Supple	RaWMS_Min	RaWMS_Max
Nombre total de transmissions	1	1	50	50	900

VII.4.5. Consommation d'énergie

La consommation d'énergie de chaque protocole a été mesurée et tracée dans la figure 73. Celle-ci montre la consommation d'énergie moyenne de chaque nœud dans la matière en fonction du nombre de lancement de la procédure de dissémination. La figure 73 montre que le protocole RaWMS consomme beaucoup plus que les autres protocoles, jusqu'à deux fois plus après 1000 phases de dissémination. Cela est dû au fait que le taux de transmission dans le réseau lors de l'utilisation USEE, DEEP et Supple est très faible (voir figure 72 de la section précédente). L'ordre trouvé dans la figure 72 relatif au nombre de transmissions est

donc le même que celui de la figure 73 sur la consommation d'énergie, c'est-à-dire le protocole DEEP est plus énergivore que le protocole USEE qui lui-même est plus énergivore que le protocole Supple.

En regard de notre contexte applicatif (tableau 8), la figure 73 nous permet de déterminer la courbe de tendance du protocole USEE, cela nous donne l'équation de courbe :

$$y = 0{,}00943782\ x + 0{,}314038$$

Pour répondre aux exigences applicatives, le protocole doit permettre au minimum un nombre d'écriture de x=11500, ce qui correspond à une énergie minimale nécessaire égale à y=109 Joules. À ce niveau de cette étude, nous concluons que le nœud doit avoir au minimum une énergie de **109 Joules** pour exécuter toutes les phases de stockage d'informations nécessaire dans le préfabriqué tout au long de son cycle de vie. Dans la suite de cette étude, nous ajouterons l'énergie nécessaire pour exécuter toutes les phases de lecture exigée par le contexte, ce qui nous donnera finalement l'énergie minimale totale que le nœud doit avoir pour notre application.

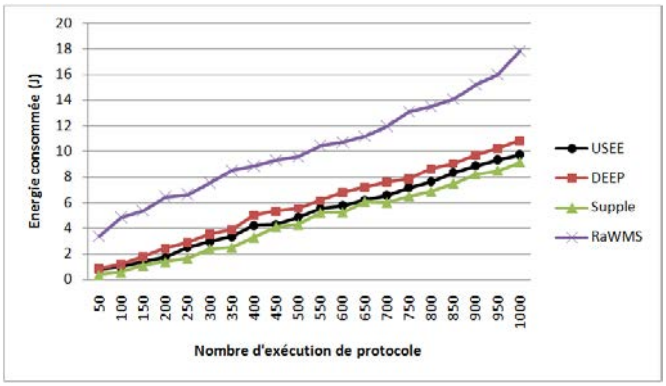


Figure 73. Consommation d'énergie moyenne d'USEE vs. Protocoles de littérature

VIII. Evaluation des performances de protocole de lecture RaWPG

Cette section présente les résultats de simulation obtenus pour la lecture d'informations dans la matière communicante en utilisant le protocole RaWPG.

VIII.1. Plan d'évaluation des performances

Dans cette étude, on exécute successivement plusieurs phases de recherche (60 fois) sur différentes zones de la matière simulée. Les résultats donnés dans cette thèse sont les moyennes de ces 60 recherches. Différentes valeurs de TTL sont testées dans cette section pour étudier les performances du protocole RaWPG. L'évaluation des performances du protocole RaWPG dans la matière communicante repose sur trois études :

- Etude probabiliste de la longueur maximale TTL du chemin de recherche à utiliser. Cette étude nous permet de déterminer la longueur maximale du chemin de recherche suivant les taux d'existence de l'information dans les voisinages produites par le protocole USEE (voir figure 62 de la section VI.3.1 de ce chapitre).

- Comparaison du protocole RaWPG avec un simple Random walk associé au Pull Gossip. Cette étude nous permet d'évaluer l'influence et les avantages d'intégrer la technique du choix d'un nœud voisin distant (bordure) et la meilleure qualité de service pour le prochain saut.
- Comparaison du protocole RaWPG avec un algorithme de recherche basé sur l'inondation de requêtes. Pour cela, nous avons choisi le protocole MARP (Sen and Ukil, 2009) présenté dans le chapitre 2. Cette étude nous permet de juger chaque technique de recherche (recherche multi-sauts et recherche par inondation) et d'analyser les avantages et les inconvénients de chacune dans le contexte de la matière communicante.

VIII.2. Critères d'évaluation des performances

Six mesures de performance sont utilisées pour évaluer le protocole RaWPG :

- *Taux de couverture* : c'est le nombre de noeuds qui ont été interrogés lors du processus de recherche d'information. Si le nombre de nœuds couverts augmente, l'information a plus de chance d'être trouvée après chaque saut.
- *Taux de succès* : La requête est transmise à plusieurs reprises sur le réseau et à chaque fois, on vérifie que le noeud maître ait bien reçu l'information. Le taux de succès présente le rapport entre le nombre de réponses reçues et le nombre de requêtes transmises dans la matière. Ce critère nous permet d'évaluer les protocoles en fonction des interférences, du bruit du canal radio, des communications asynchrones, etc.
- *Délai de lecture* : c'est le temps écoulé sur le nœud maitre entre l'instant du début de la diffusion de la requête et l'instant de réception du message de réponse correspondant.
- *Taux de duplication* : cette mesure présente le nombre de messages de réponse qui sont renvoyés au nœud maître, suite à la transmission d'une requête de lecture dans la matière. Un taux de duplication élevé est un inconvénient majeur du protocole dans les réseaux de capteurs sans fil.
- *Nombre de communications* : c'est le nombre de transmissions de message (requête et réponse) dans le réseau pendant le processus de recherche. Comme cela a été discuté dans ce chapitre, cette mesure est un indicateur fort de la consommation d'énergie dans le réseau de capteurs.
- *Consommation d'énergie* : c'est l'énergie moyenne consommée par chaque nœud de capteurs dans la matière après plusieurs éxecutions du protocole de recherche d'information dans la matière.

VIII.3. Etude de la longueur maximale de chemin de recherche TTL

Dans le processus de dissémination USEE développé, l'information est stockée dans chaque nœud de capteurs avec une probabilité de stockage *P*. Ce processus a donné une probabilité

d'existence de l'information dans un voisinage p(k) (voir section VI.3.1 de ce chapitre). Soit *p* cette probabilité d'existence de l'information dans un voisinage du nœud dans le préfabriqué.

Le protocole RaWPG consiste à faire différents sauts dans le réseau de capteurs enfoui dans la matière pour trouver l'information comme le montre la figure 74. À chaque saut, la requête est diffusée dans le voisinage. À chaque diffusion au niveau d'un saut, il y a deux réponses possibles : soit le voisinage contient l'information dans la mémoire de l'un des nœuds voisins, soit aucun nœud de voisinage ne contient l'information.

Donc, à chaque diffusion dans le voisinage d'un saut, on a deux réponses possibles : un succès (information trouvée dans le voisinage) ou un échec (l'information n'existe pas dans le voisinage). Ce processus de recherche représente une *épreuve de Bernoulli* où tout lancer ne possède que deux issues possibles, que l'on appelle *succès* et *échec*.

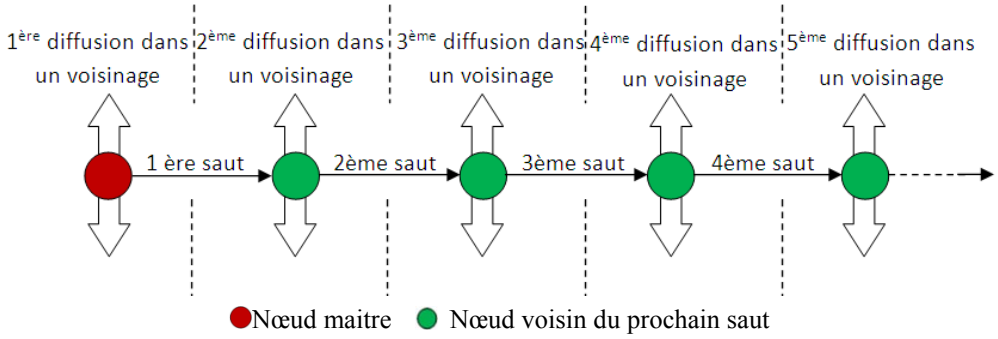


Figure 74. Recherche par diffusion multi-sauts dans le réseau enfoui dans la matière

Le processus de recherche d'information représente une expérience aléatoire consistant à répéter *m* fois (*m* sauts) de façon indépendante une épreuve de Bernoulli de paramètre *p*. Ce qui nous ramène à présenter la recherche d'information dans une matière communicante en un schéma de Bernoulli de paramètres *m* et *p*. Ce schéma de Bernoulli peut être modélisé par un arbre qui aurait deux branches initiales (succès-échec) comme le montre la figure 75, et chacune de ces branches donne naissance à deux nouvelles branches (succès-échec). À chaque épreuve, le nombre de branches est doublé. Chaque branche représente une diffusion au niveau d'un saut dans le processus de recherche RaWPG dans le réseau de capteurs.

Une succession de *m* épreuves de Bernoulli indépendantes ($m \in N^*$) permet la construction d'une variable aléatoire S_m comptant le nombre de succès au bout de *m* épreuves de Bernoulli. Cette variable aléatoire a pour loi de probabilité la *loi binomiale* de paramètres *m* et *p* notée *B(m,p)*. S_m est défini de la manière suivante :

$$S_m = \sum_{i=1}^{m} x_i$$

Suivant la loi binomiale, pour tout $0 \leq k \leq m$ on a :

$$P(S_m = k) = \binom{m}{k} p^k q^{m-k}$$

Avec :

k : le nombre de fois d'avoir un succès.

m : le nombre d'épreuves exécutées.

p : la probabilité d'avoir un succès à une épreuve.

q : la probabilité d'avoir un échec à une épreuve.

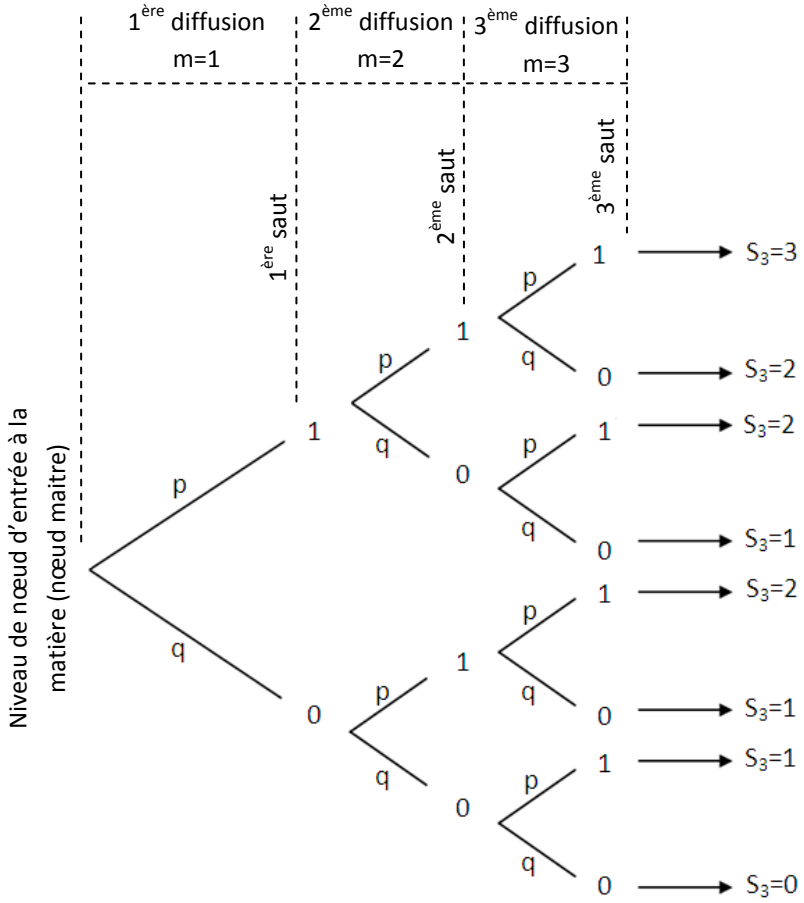


Figure 75. Modélisation du schéma de Bernoulli par arbre de recherche d'une information

Soit $P(S_m \geq 1)$ la probabilité de trouver l'information au moins une seule fois après *m* sauts. On a :

$$P(S_m \geq 1) = \sum_{k=1}^{m} P(S_m = k)$$

D'autre part on obtient, en supprimant le cas où on ne trouve pas l'information après *m* sauts :

$$\sum_{k=0}^{m} P(S_m = k) = P(S_m = 0) + \sum_{k=1}^{m} P(S_m = k)$$

Suivant la fonction de répartition d'une variable aléatoire S_m qui suit une loi binomiale *B(m,p)*, on a :

$$\sum_{k=0}^{m} P(S_m = k) = 1$$

D'où :

$$P(S_m \geq 1) = 1 - P(S_m = 0)$$

$$P(S_m \geq 1) = 1 - \binom{m}{0} p^0 q^m$$

$$P(S_m \geq 1) = 1 - q^m$$

$$P(S_m \geq 1) = 1 - (1-p)^m$$

$$1 - P(S_m \geq 1) = (1-p)^m$$

$$\log(1 - P(S_m \geq 1)) = m \times \log(1-p)$$

Donc pour tout $P(S_m \geq 1) \neq 1$ et $p \neq 1$ on a :

$$m = \frac{\log(1 - P(S_m \geq 1))}{\log(1-p)}$$

Ce résultat permet de limiter la zone de recherche autour du nœud maitre. Afin d'être à peu près certain de trouver l'information après *m* sauts, on fixe $P(S_m \geq 1) = 0.99$. D'où pour tout $0<p<1$ on a :

$$m = \frac{-3}{\log(1-p)}$$

La variation de *m* en fonction de *p* est présentée sur la figure 76. Pour une probabilité d'existence d'information dans le voisinage *p* donnée, la figure 76 nous permet de déterminer la valeur *m* de TTL maximale à utiliser dans le protocole de lecture RaWPG. La valeur TTL déterminée permet de trouver l'information avec une probabilité de 0.99.

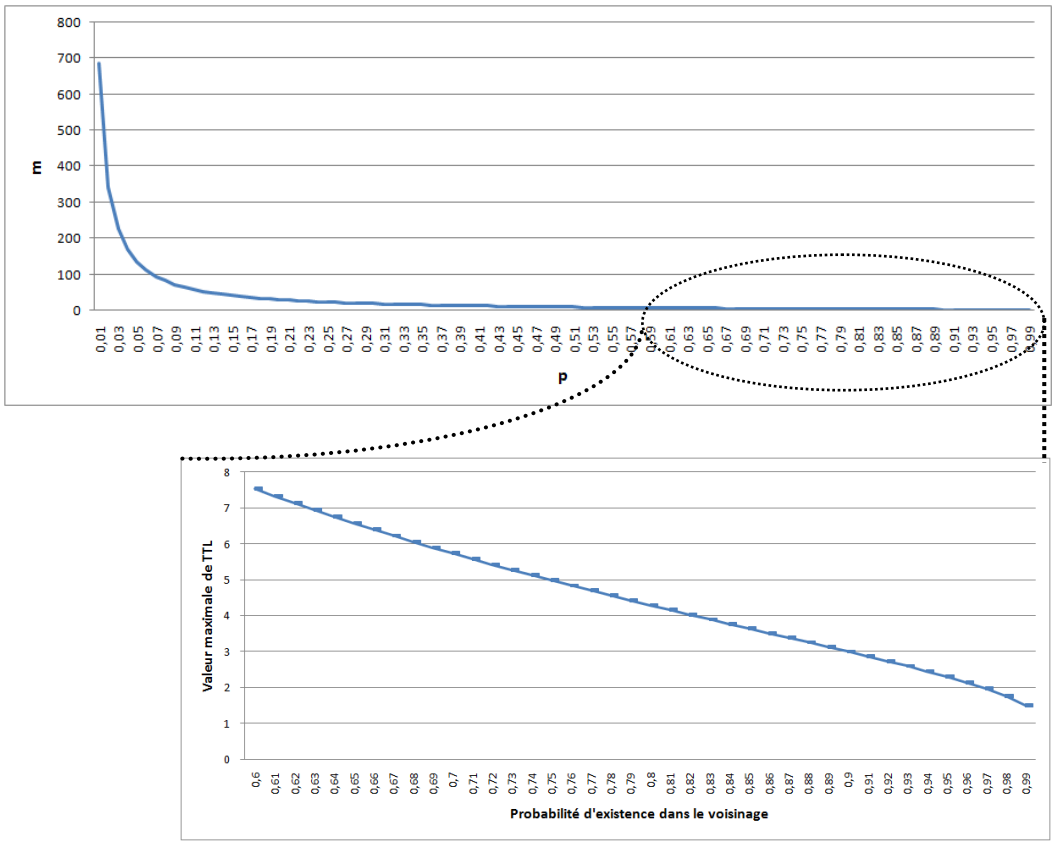


Figure 76. Longueur maximale de TTL suivant la probabilité d'existence dans les voisinages

La probabilité d'existence de l'information dans les voisinages est illustrée par la figure 62. Pour le cas étudié dans ce chapitre (n=25), la probabilité d'existence dans les voisinages varie entre 0.6 et 1 pour toutes les probabilités de stockage. En se basant sur la relation entre la probabilité d'existence dans les voisinages et la longueur de chemin maximale présentée dans la figure 76, nous fixons la longueur maximale TTL à 8 dans le reste de cette étude. Ainsi, la requête fera au maximum 8 sauts pour chercher une donnée dans la matière. Suivant la probabilité de stockage, une information sera trouvée après n'importe qu'elle valeur de TTL entre 1 et 8. Afin de bien évaluer les performances du protocole RaWPG dans tous les cas possibles, nous procédons comme suit :

- Nous disséminons des informations avec toutes les probabilités de stockage (0.1, 0.2, …, 0.9 et 1) en utilisant le protocole USEE.
- Pour chaque information nous utilisons la valeur TTL correspondante qui permet sa lecture. La valeur TTL est déterminée par la relation entre la probabilité de stockage et le taux d'existence dans les voisinages pour n=25 présenté dans la figure 62. Par exemple, pour une information stockée avec une probabilité 0.5, le taux d'existence dans les voisinages est 0.9 dans la figure 62. Ce taux d'existence correspond à la valeur TTL=3 dans la figure 76. De cette manière, il est possible d'évaluer les performances du protocole RaWPG pour lire les informations pour toutes les probabilités de stockage possibles en utilisant la longueur de chemin TTL (entre 1 et TTL maximale) correspondante à chaque probabilité.

Dans la suite de ce chapitre, les performances de RaWPG sont comparées à un simple Random Walk+Pull Gossip et à MARP pour les différentes longueurs de chemin entre 1 et TTL maximale.

VIII.4. RaWPG vs. Simple Random Walk + Pull Gossip

La comparaison du protocole RaWPG avec un simple Random Walk+Pull Gossip est faite en terme de nombre de nœuds couverts dans la phase de recherche (nœuds interrogés). Puis, on évalue l'algorithme en terme d'efficacité de réception des réponses. Finalement, une étude du délai est présentée.

VIII.4.1. Taux de couverture

La figure 77 donne la différence entre le nombre de nœuds interrogés en utilisant le processus de recherche du protocole RaWPG et celui d'un simple Random Walk couplé à un mécanisme Pull Gossip. Les résultats montrent que la technique de choix des nœuds les plus distants est une stratégie gagnante comparée à un processus de recherche à marche aléatoire. D'après la figure 77, le protocole RaWPG permet de découvrir plus de nœuds en fonction du nombre de sauts et jusqu'à plus de 30% de nœuds supplémentaires pour un TTL égal à 8. Ce

résultat permet d'accroitre la chance de trouver l'information recherchée, en utilisant le même nombre de sauts que si l'on avait mis en place un simple Random Walk et Pull Gossip.

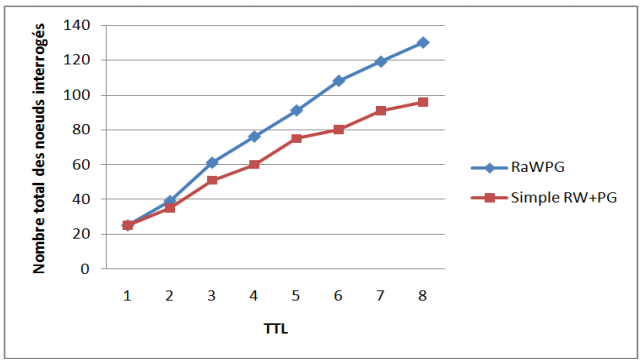


Figure 77. Taux de couverture de RaWPG vs. Simple Random Walk+ Pull Gossip

VIII.4.2. Taux de succès

Le processus de recherche consiste à diffuser une requête par le nœud maitre et attendre la réception d'un message de réponse de son nœud de prochain saut dans le chemin de recherche. La figure 78 montre le taux de succès moyen de réception de réponse suite à plusieurs opérations de lecture dans différentes zones de la matière. Cette étude nous permet d'évaluer la fiabilité des chemins de recherche. Le chemin est dit fiable s'il permet d'assurer l'acheminement d'une requête et de retourner une réponse trouvée. La figure 78 indique bien que le protocole RaWPG permet un taux de succès plus élevé que lorsque l'on utilise un simple Random Walk et Pull Gossip. Le taux diminue légèrement avec l'augmentation du nombre de sauts mais il reste toujours supérieur à 90%. Ce résultat confirme la fiabilité de chemin suivie par la requête. Le choix d'un nœud reposant sur la qualité de service permet d'avoir une connexion plus fiable réduisant ainsi les pertes de messages causées par les interférences et le bruit du canal radio. Par opposition, le caractère aléatoire du choix des nœuds de l'algorithme Random Walk+Pull Gossip peut conduire à sélectionner des chemins moins fiables.

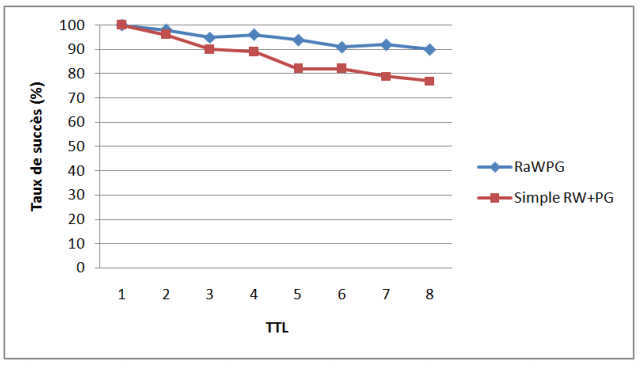


Figure 78. Taux de succès de RaWPG vs. Simple Random Walk+ Pull Gossip

VIII.4.3. Délai de lecture

Le délai de réception d'une réponse par le nœud maitre est un indicateur important dans l'évaluation du protocole de recherche d'information dans les réseaux de capteurs sans fil. Il

représente le temps écoulé entre le début de la transmission d'une requête dans le réseau et la réception du message de réponse par le nœud maître. La figure 79 représente les délais moyens pour le protocole RaWPG et pour un simple Random Walk+Pull Gossip en fonction de plusieurs TTL. La figure 79 montre que les deux protocoles ont presque les mêmes délais. En effet, le protocole RaWPG utilise des temps d'attente à chaque saut pour interroger les voisins. L'attente et le traitement des messages FAR_INFO entraînent des retards supplémentaires, mais cela est compensé par le fait que la requête dans RaWPG a plus de chance de trouver l'information (sélection du voisin le plus éloigné) que dans le cas où on utilise un simple Random Walk+Pull Gossip.

De plus, par rapport à notre contexte applicatif (tableau 8), la contrainte de délai sur la lecture des informations dans le préfabriqué béton doit être inférieure à 5 secondes. Les résultats dans la figure 79 répondent parfaitement aux exigences puisque le protocole de lecture RaWPG est en dessous de 1 seconde pour toutes les longueurs de chemins.

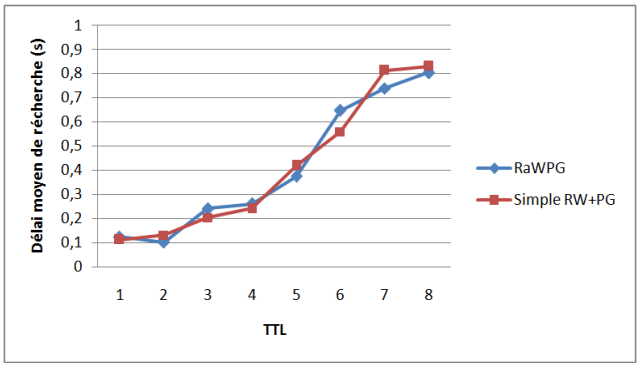
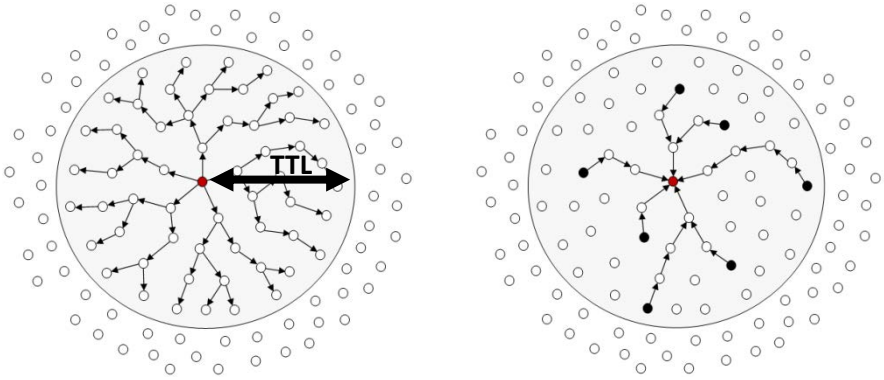


Figure 79. Délai de lecture de RaWPG vs. Simple Random Walk+ Pull Gossip

VIII.5. RaWPG vs. MARP

Cette étude compare les performances du protocole RaWPG avec le protocole MARP. MARP est basé sur l'inondation d'une requête dans le réseau pour trouver une information. On utilise le champ TTL pour limiter la zone d'inondation de ce protocole (voir figure 80(a)). L'information demandée est envoyée par les nœuds sur le chemin inverse parcouru par la requête. Comme l'information est répliquée dans plusieurs nœuds de la matière, plusieurs sources peuvent donc répondre (voir figure 80(b)). Dans la suite de cette section, on évalue les performances et les avantages/inconvénients du protocole RaWPG par rapport au protocole MARP. Les protocoles sont évalués en termes de taux de succès, de taux de duplication, de taux de communications et de consommation d'énergie.



(a) (b)

Figure 80. Fonctionnement du protocole de recherche d'information MARP

VIII.5.1. Taux de succès

La figure 81 donne le taux moyen de réception d'une réponse pour les protocoles RaWPG et MARP. MARP est meilleur que le protocole RaWPG avec un taux de succès toujours supérieur à 95% alors que le RaWPG se situe au dessus de 90%. Cela provient du fait que le protocole MARP utilise la technique d'inondation de la requête pour récupérer les informations demandées. Plusieurs sources d'information répondent. D'où, le message de réponse est envoyé vers le nœud maître à travers de nombreux chemins, et si la transmission échoue sur un chemin, l'autre va réussir ce qui explique le taux de succès élevé pour le protocole MARP. Cette technique de recherche n'est pas implémentée dans le protocole RaWPG. Cependant, nous avons utilisé une sélection de chemin à haute qualité de service afin d'améliorer la fiabilité de transmission ce qui augmente au maximum le taux de succès.

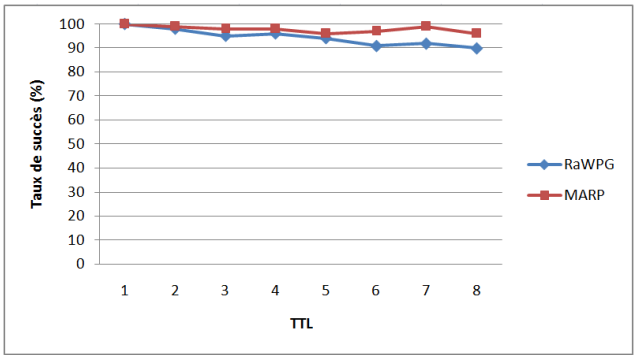


Figure 81. Taux de succès de RaWPG vs. MARP

VIII.5.2. Taux de duplication

Dans notre contexte de la matière communicante, les informations sont répliquées dans de nombreux nœuds du réseau. Si une requête est inondée dans le réseau, tous les nœuds qui stockent l'information répondent par un message de réponse. Pour cela, la figure 82 montre que MARP a un taux de duplication de réponse élevé. Evidemment, si la valeur de TTL augmente, la zone d'inondation devient plus importante, puisque la requête atteint plus de nœuds (augmentation de taux de duplication).

Cependant, en utilisant la marche saut par saut du protocole RaWPG pour acheminer la requête, le processus de recherche s'arrête au premier nœud qui stocke l'information. Ainsi, le nœud maître reçoit une seule fois la réponse finale quelle que soit la valeur de TTL comme l'illustre la figure 82. Cela permet d'optimiser l'utilisation des ressources du réseau et plus spécifiquement du nœud maitre au cours d'un processus de recherche.

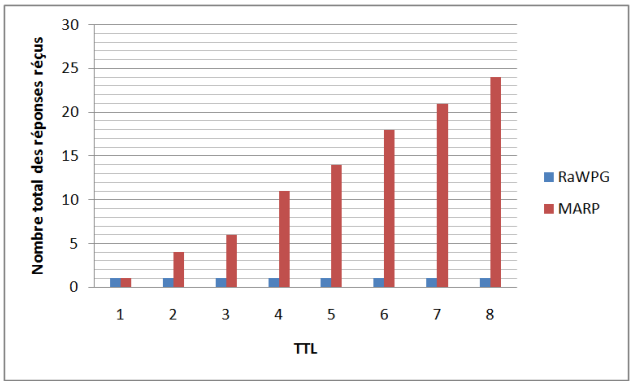


Figure 82. Taux de duplication de RaWPG vs. MARP

VIII.5.3. Nombre de communications

La figure 83 donne le nombre total de messages transmis dans le réseau pour les protocoles RaWPG et MARP en fonction de plusieurs valeurs de TTL. Tout d'abord, le nombre de communications de MARP est élevé à cause de la technique d'inondation. Comme indiqué précédemment, le protocole MARP a un taux de duplication de réponse élevé surtout pour de grandes valeurs de TTL. Cela peut conduire à la congestion du réseau, et à des interférences entre les communications.

Comparativement à la technique d'inondation proposée dans le protocole MARP, la technique saut par saut du protocole RaWPG limite le nombre de messages envoyés sur le réseau même si la valeur de TTL augmente. Le gain est conséquent puisque le nombre de messages est divisé par deux à partir d'un TTL supérieur à 2. Comme la communication est l'activité principale responsable de la consommation d'énergie dans les réseaux de capteurs sans fil, ces résultats vont avoir une forte incidence sur l'efficience énergétique du protocole RaWPG par rapport au protocole MARP.

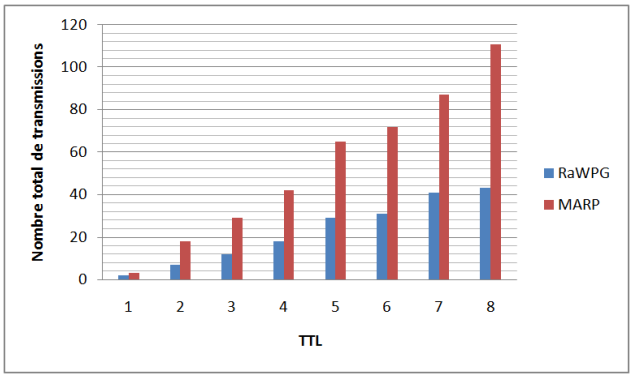


Figure 83. Nombre de communications de RaWPG vs. MARP

VIII.5.4. Consommation d'énergie

La performance de la consommation d'énergie de chaque protocole est donnée sur la figure 84. La figure 84 présente la consommation d'énergie moyenne de chaque nœud dans la matière en fonction du nombre d'exécution du protocole. Comme attendu par rapport aux résultats de la section précédente sur l'analyse du nombre de messages générés par les deux protocoles, le protocole RaWPG consomme beaucoup moins d'énergie que le protocole MARP. L'écart de consommation entre ces deux protocoles est de plus en plus grand au fur et à mesure que l'on lance des requêtes de lecture.

Relativement au notre contexte applicatif (tableau 8), la figure 84 nous permet de déterminer la courbe de tendance du protocole RaWPG, ayant pour équation :

$$y = 0{,}009864048\ x$$

Pour répondre aux exigences applicatives, le nombre de lectures est au minimum x=11553, cela correspond à une énergie minimale nécessaire égale à y=**114 Joules**. Nous avons trouvé précédemment dans cette étude que le nœud doit avoir au minimum une énergie de 109 Joules pour exécuter toutes les phases de stockage (voir section VII.4.5. de ce chapitre). En ajoutant l'énergie de lecture nécessaire, nous concluons que le nœud doit avoir une énergie minimale totale de **223 Joules** pour assurer toutes les phases de stockage et de lecture d'informations nécessaire dans le préfabriqué tout au long de son cycle de vie.

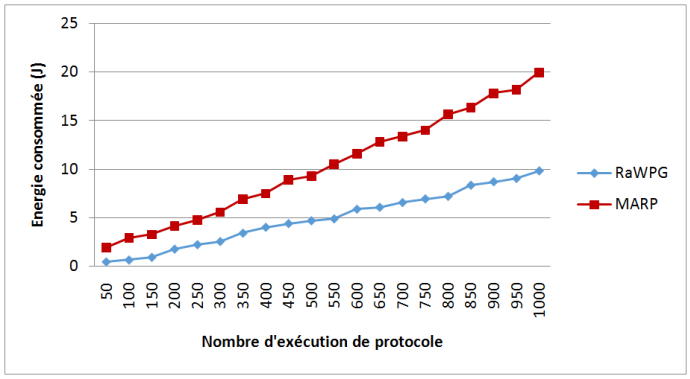


Figure 84. Consommation d'énergie moyenne de RaWPG vs. MARP en fonction du nombre de lecture

IX. Synthèse et discussion

L'impact de plusieurs paramètres sur la performance des protocoles proposés dans cette thèse nommés USEE (Uniform Storage and Energy Efficient data dissemination protocol) et RaWPG (Random Walk and Pull Gossip data retrieval protocol) ont été étudié par simulation. Nous avons choisi le simulateur Castalia sous la plateforme OMNeT++ car il offre un environnement de simulation proche de la réalité pour les réseaux de capteurs sans fil (Pediaditakis et al., 2010) (Rastegarnia et al., 2011). En outre, nous avons implémenté le micro-nœud de capteurs SensorCube dans ce simulateur. En utilisant ce micro-nœud, on a testé par simulation un préfabriqué enfouissant des centaines de nœuds. Sur la base des

résultats obtenus par simulations dans ce chapitre, les conclusions ci-dessous peuvent être tirées.

- Etude de la densité des nœuds de capteurs dans un préfabriqué :

Nous avons étudié la densité des nœuds dans un préfabriqué pour les meilleures performances d'une dissémination probabiliste uniforme et une lecture d'information efficace. Deux études ont été réalisées : une étude théorique en utilisant la loi de probabilité hypergéométrique et une étude par simulation. Les nœuds de capteurs sont uniformément déployés dans le préfabriqué. Pour l'étude de densité des nœuds vis-à-vis de l'uniformité de dissémination dans USEE, nous avons donc supposé que l'information existe dans tous les endroits du préfabriqué, si l'information existe dans le voisinage de chaque nœud dans le réseau de capteurs. Les résultats de simulation ont démontré que plus on augmente le nombre de nœuds dans le voisinage, plus la probabilité d'existence de l'information dans chaque voisinage augmente, ce qui augmente la probabilité de trouver l'information dans toutes les endroits du préfabriqué.

Pour l'étude de densité des nœuds vis-à-vis l'efficacité de la sélection des nœuds de bordure dans RaWPG, nous avons analysé l'influence de la variation du nombre de nœuds dans un voisinage sur l'efficacité de la stratégie de sélection. Les résultats de simulation ont démontré que l'algorithme fonctionne parfaitement pour une grande densité dans le voisinage (exemple : 25, 36 et 45) d'un nœud dans le préfabriqué.

- Stockage d'information dans la matière communicante :

Le protocole de dissémination et stockage USEE a été étudié en analysant l'effet du mécanisme de réplication contrôlé NS. Afin d'analyser cet effet, nous avons supprimé le mécanisme de contrôle NS du protocole USEE et comparé les résultats avec l'algorithme initial d'USEE en termes de taux de réplication et de capacité de stockage du réseau, à la suite de plusieurs opérations de dissémination. Cette étude a montré que le mécanisme NS réduit le nombre de nœuds stockant l'information. Le protocole USEE pourrait atteindre 27000 opérations de stockage dans un réseau formé de 2500 micro-nœuds SensorCube qui ont chacun une mémoire de 120 Ko. Par contre, USEE sans NS n'autorise que 7000 opérations de stockage pour le même réseau.

La deuxième partie d'évaluation des performances d'USEE a été faite par comparaison avec d'autres protocoles de dissémination uniforme de la littérature: DEEP, Supple et RaWMS.

Ces protocoles de la littérature ont été choisis, suite à une étude approfondie de toutes les techniques et tous les mécanismes existants actuellement. Chaque protocole utilise des mécanismes différents : DEEP est basé sur l'inondation et le stockage probabiliste, Supple repose sur la transmission hiérarchique et le stockage probabiliste, et RaWMS s'appuie sur la transmission à marche aléatoire et le stockage par compteur de sauts. Ces comparaisons

permettent d'analyser les avantages et les inconvénients du protocole USEE avec d'autres techniques utilisées dans la littérature. Ces protocoles ont été comparés en termes d'uniformité, de taux de réplication, de délai, de nombre de communication et de consommation d'énergie.

Cette étude a montré qu'un protocole basé sur le stockage probabiliste (USEE, DEEP et Supple) a une dissémination des données plus uniforme qu'un protocole basé sur le stockage par comptage de nombre de sauts (RaWMS). En outre, les protocoles utilisant l'inondation (USEE et DEEP) permettent des uniformités plus élevées par rapport à la transmission hiérarchique (Supple) et à la marche aléatoire (RaWMS). L'inondation dans un réseau de capteurs assure la réception de messages sur tous les nœuds de la matière, et cela a un effet positif sur l'uniformité de la réplication. Ainsi, une technique de dissémination basée à la fois sur l'inondation et le stockage probabiliste dans chaque nœud (USEE et DEEP), assure que les informations soient répliquées dans la matière de manière uniforme et selon leur niveau d'importance.

De plus, l'inondation counter-based adoptée par notre protocole USEE consomme moins d'énergie par rapport à l'inondation probabiliste du protocole DEEP. Cela est dû au fait que l'inondation counter-based réduit le nombre de retransmissions du message, tout en assurant sa réception par tous les nœuds du réseau.

En outre, le protocole USEE est le seul protocole qui utilise un mécanisme pour contrôler le taux de réplication dans chaque voisinage de la matière. L'étude sur le taux de réplication dans le réseau a montré que le protocole USEE limite la surcharge des mémoires comparativement aux autres protocoles, et maintient une densité de stockage contrôlée dans la matière pour tous les niveaux d'importance d'information. Pour les protocoles DEEP, Supple et RaWMS, le taux de réplication est linéaire surtout pour des niveaux élevés d'importance ce qui conduit à saturer la mémoire des nœuds, après plusieurs processus de stockage dans la matière.

Le temps écoulé pour accomplir un processus de dissémination d'une information dans la matière communicante a aussi été étudié. Le délai d'attente *RAD* (Random Assessment Delay) à chaque saut, inclus dans le mécanisme d'inondation counter-based d'USEE, a montré son influence négative sur l'augmentation du délai global de dissémination dans la matière. Le protocole DEEP a le plus faible délai car il utilise l'inondation probabiliste. Dans une inondation probabiliste, chaque nœud diffuse directement le message avec une probabilité fixée ce qui accélère le processus de dissémination. Malgré cela, le protocole USEE a un faible délai par rapport aux protocoles de dissémination hiérarchique Supple et de dissémination par marche aléatoire RaWMS.

- Lecture d'information dans la matière communicante :

Le protocole de lecture RaWPG est basé sur un algorithme de sélection des nœuds voisins distants. Le lancement de l'algorithme de sélection des nœuds voisins distants est une phase primordiale afin de préparer les nœuds et le réseau pour l'exécution du processus de recherche de RaWPG. Deux comparaisons ont été étudiées pour évaluer les performances de RaWPG.

Tout d'abord, le protocole RaWPG a été comparé à un simple processus Random Walk couplé à un Pull Gossip. Les deux protocoles ont été analysés en termes de nombre total de nœuds couverts, de taux de succès et de délai de lecture. Cette comparaison a montré que le protocole RaWPG couvre et interroge un nombre de nœuds plus grand qu'un simple Random Walk + Pull Gossip. Cela s'explique par l'effet de l'algorithme de sélection des nœuds voisins distants adopté par le protocole RaWPG qui consiste à sélectionner le voisin le plus éloigné comme prochain saut, ce qui permet de couvrir le maximum de nouveaux nœuds à chaque saut. L'augmentation du nombre de nœuds couverts permet d'avoir plus de chance de trouver l'information dans chaque saut dans un processus de recherche multi-sauts. En outre, le protocole RaWPG est meilleur qu'un simple Random Walk + Pull Gossip en taux de succès de recherche d'information grâce à la fonction de coût de liaison. Contrairement à la sélection du chemin aléatoire de Random Walk + Pull Gossip, les messages de requête et de réponse dans le protocole RaWPG sont transmis par des chemins fiables à haute qualité de service. Cependant, le choix du voisin qui permet un service de haute qualité impose un délai d'attente dans chaque saut pour traiter les messages FAR_INFO, ce qui augmente le délai global du processus de lecture de RaWPG.

Ensuite, le protocole RaWPG a été comparé au protocole MARP. Cette comparaison a montré que le protocole RaWPG a plusieurs avantages par rapport au protocole MARP dans le contexte applicatif de cette thèse, où les données sont uniformément répliquées dans le réseau. Cela signifie que l'information est stockée dans plusieurs nœuds à l'intérieur de la matière. Si une requête inonde le réseau, tous les nœuds stockant l'information envoient des messages de réponse, ce qui conduit à un taux de duplication de réponse élevé. Pour cette raison, le protocole MARP a un taux de duplication élevé par rapport au protocole RaWPG. En effet, RaWPG est un protocole de transmission multi-sauts sur un seul chemin. Quelle que soit la profondeur de ce chemin dans la matière, le protocole RaWPG s'arrête à la première réponse trouvée. Donc, aucune duplication de réponse n'est faite, ce qui optimise le taux de communication et l'énergie consommée au cours de la recherche d'une information. Cependant, le protocole MARP tire avantage de la duplication de réponse pour augmenter le taux de succès. En effet, si plusieurs messages de réponse sont envoyés vers le nœud maître à travers différents chemins, cela renforce la probabilité d'acheminer correctement le message. Mais cela se fait notamment au détriment de la consommation d'énergie du réseau.

- Consommation d'énergie et durée de vie du préfabriqué en béton :

Les expériences menées par simulation ont été réalisées par rapport à un contexte applicatif précis qui est le suivi d'un préfabriqué tout au long de son cycle de vie. Les résultats de simulation montrent l'intérêt des mécanismes d'optimisation spécifiés dans les protocoles USEE et RaWPG qui répondent parfaitement aux exigences applicatives comme le nombre d'informations pouvant être stockées ou lues durant toute la vie du bâtiment ou encore par rapport à des contraintes temporelles sur les phases de lecture et d'écriture.

Cette étude a permis aussi de dimensionner l'autonomie nécessaire de chaque nœud de capteurs pour l'ensemble du cycle de vie de notre application avec une valeur d'énergie minimale de 223 Joules. En ajoutant les taux d'énergie nécessaires pour l'acquisition de nouvelles informations par le capteur et le taux d'énergie consommé en mode non actif, le nœud SensorCube (tension d'alimentation = 2.4 V) doit être alimenté par une ressource d'énergie (par exemple une pile) de capacité minimale de 350mAh comme illustré par le tableau 12, afin d'assurer une durée de vie supérieure à 14 ans (durée de BOL + MOL + EOL du préfabriqué en béton).

Tableau 12. Durée de vie de nœud vs. Capacité de batterie

Capacité de batterie	50mAh	150mAh	250mAh	350mAh	450mAh	750mAh	1000mAh
Durée de vie de nœud	1,1 an	5,6 ans	10,2 ans	14,8 ans	19,4 ans	33 ans	44,5 ans

Tout au long du cycle de vie du préfabriqué, deux classes d'informations sont stockées et lues comme cela a été présenté précédemment :

- *Les données externes* : qui correspondent aux données issues de la base de donnée et stockés dans les nœuds telles que l'identifiant du préfabriqué, sa longueur, sa hauteur, la référence du transporteur, l'identité du chantier, la fiche manutention, le calepinage, la date d'opération de maintenance, etc.
- *Les données internes* : qui sont les données générées par les capteurs afin de surveiller le béton : température, humidité et détection de fissures.

On peut voir dans le tableau 13, que le nombre d'opérations de stockage et de lecture de données internes est plus grand que pour celui des données externes. Les données internes sont stockées 11286 fois et lues 10964 fois durant tout le cycle de vie du préfabriqué. Tandis que les données externes sont stockées 214 fois et lues 489 fois. Cela conduit donc à un taux de consommation d'énergie plus élevé pour la gestion des données internes avec 214 Joules pour le stockage et la lecture des données internes et 9 Joules pour les opérations sur les données externes. En conclusion, les données internes consomment 95% de la totalité d'énergie nécessaire pour le stockage et la lecture des informations durant tout le cycle de vie

du préfabriqué. Tandis que les données externes ne consomment que 5% de l'énergie total (voir figure 85).

Tableau 13. Consommation d'énergie des données internes et externes

Stockage		Lecture	
Nombre total d'opérations = 11500		**Nombre total d'opérations = 11553**	
Données internes	**Données externes**	**Données internes**	**Données externes**
11286 opérations	214 opérations	10964 opérations	589 opérations
106 Joules	3 Joules	108 Joules	6 Joules

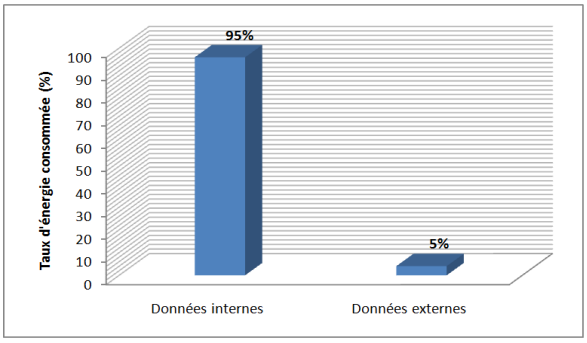


Figure 85. Taux de consommation d'énergie pour les données internes et externes par rapport à l'énergie totale consommé en stockage et lecture durant tout le cycle de vie du préfabriqué

Dans cette thèse, l'étude de la consommation d'énergie a considéré les phases de stockage, de lecture, de capture d'informations environnementales ainsi que la phase d'inactivité du nœud SensorCube. Le taux de consommation d'énergie va donc dépendre du nombre d'opérations à exécuter mais va aussi être fonction de la manière de gérer les états de veille du réseau entre chaque opération. La figure 86 montre ces phases que se succèdent. Les consommations données dans les tableaux 12 et 13 considèrent que le nœud SensorCube consomme en mode veille 2.5 µA (voir tableau 10). Cependant, cette valeur ne prend pas en compte le vieillissement de la batterie qui va induire un déchargement prématuré de celle-ci. Le but de nos calculs sur la consommation d'énergie était d'évaluer les performances énergétiques des protocoles USEE et RaWPG indépendamment des caractéristiques intrinsèques des batteries. Un autre facteur qui devrait aussi être étudié pour avoir une évaluation globale de la consommation d'énergie de notre système est l'influence de la couche MAC qui sera utilisée en dessous de nos deux protocoles. Dans le simulateur, nous avons utilisé la couche T-MAC qui n'est pas un modèle optimisé pour la consommation d'énergie (Song, 2013) mais qui nous a permis d'une part d'implémenter nos protocoles dans le simulateur et d'autre part d'avoir une référence commune (même couche MAC) pour comparer nos propositions avec d'autres travaux de recherche.

Dans la suite de nos travaux, il serait intéressant d'étudier des protocoles MAC à faible consommation d'énergie, c'est-à-dire qui sont capables d'optimiser les phases d'activité et les phases d'endormissement. Nous pouvons distinguer deux approches :

- Le réveil synchrone pour le stockage des données internes, avec des fréquences de l'ordre une fois par heure pendant 14 jours en maturométrie au cours du BOL, et une fois par jour durant le MOL. Dans ce cas, les horloges des nœuds sont synchronisées pour qu'ils se réveillent en même instant. Par exemple, suite au réveil durant le MOL, chaque nœud dans le béton dissémine les données dans le réseau uniquement s'il détecte une variable ambiante anormale (température ou humidité anormale, détection de fissure). Cette phase peut être assurée par un protocole MAC synchrone à base d'échanges de calendrier de réveil entre les nœuds de capteurs, comme S-MAC, T-MAC, D-MAC, Q-MAC, R-MAC ou LO-MAC (Thérèse, 2016) (ElFatni et al., 2011).
- Le réveil des nœuds lorsque l'utilisateur souhaite stocker des informations externes ou lire des informations externes ou internes. Dans ce cas, pour réveiller les nœuds, l'utilisation d'un protocole MAC asynchrone aura un impact négatif sur l'autonomie du réseau puisque dans ce type de protocole MAC, chaque nœud écoute le canal radio périodiquement (Musilek et al., 2015). En fait, le nombre d'opérations total de stockage et de lecture dans notre application est très faible (inférieur à 20000) par rapport à durée de vie total du préfabriqué (14 ans). Donc, l'utilisation d'un protocole MAC asynchrone pour réveiller périodiquement les nœuds afin d'exécuter un nombre faible d'opérations, semble non efficace et va générer un gaspillage d'énergie. Pour cette raison, nous envisageons plutôt des techniques de réveil par récepteur de signal radio à faible puissance (très faible taux de consommation d'énergie) qui transmet un courant d'interruption au microcontrôleur pour activer le nœud (Jelicic et al., 2012) (Magno et al., 2014). Ces récepteurs fonctionnent par des signaux radio de fréquences différentes de celles utilisées par le module radio du nœud de capteurs (par exemple 1.9 Ghz, 868 Mhz, 433 Mhz et 150 Khz) et de débit très faible (par exemple 50 Kbps, 10 Kbps et 1 Kbps). Deux classes de récepteurs existent : les récepteurs de réveil actifs et les récepteurs de réveil passifs. Dans la première classe, le récepteur se met en écoute périodique au canal radio dédié au réveil pour détecter les signaux de réveil des nœuds voisins. Le récepteur de réveil actif le moins énergivore dans la littérature consomme 270 nW (Marinkovic and Popovici, 2011) tandis que le plus énergivore consomme 1.1 mW (Shih et al., 2011). Pour la classe de récepteurs de réveil passif, ils utilisent des circuits intégrés qui ne consomment pas l'énergie de la batterie du nœud. En effet, ces circuits récupèrent l'énergie nécessaire pour leur fonctionnement à partir du signal de réveil reçu (Musilek et al., 2015). Pour réveiller le nœud, une période de temps de réception d'un signal radio est nécessaire pour générer une quantité d'énergie non négligeable. Par exemple dans (Gu and Stankovic, 2005), la transmission d'un signal de fréquence 433 Mhz durant 5 ms est nécessaire pour réveiller un nœud situé à une distance maximale de 30m. En contrepartie,

cela augmente le délai de l'exécution des protocoles de communication à chaque saut dans le réseau de capteurs.

En conclusion, la technique d'interruption de microcontrôleur par récepteur de signal radio de faible puissance nécessite évidement des modifications matérielles du nœud SensorCube. Mais, un module dédié au réveil du nœud par signal radio pourrait être ajouté au SensorCube puisque son architecture est modulaire et extensible comme cela est expliqué au début de ce chapitre.

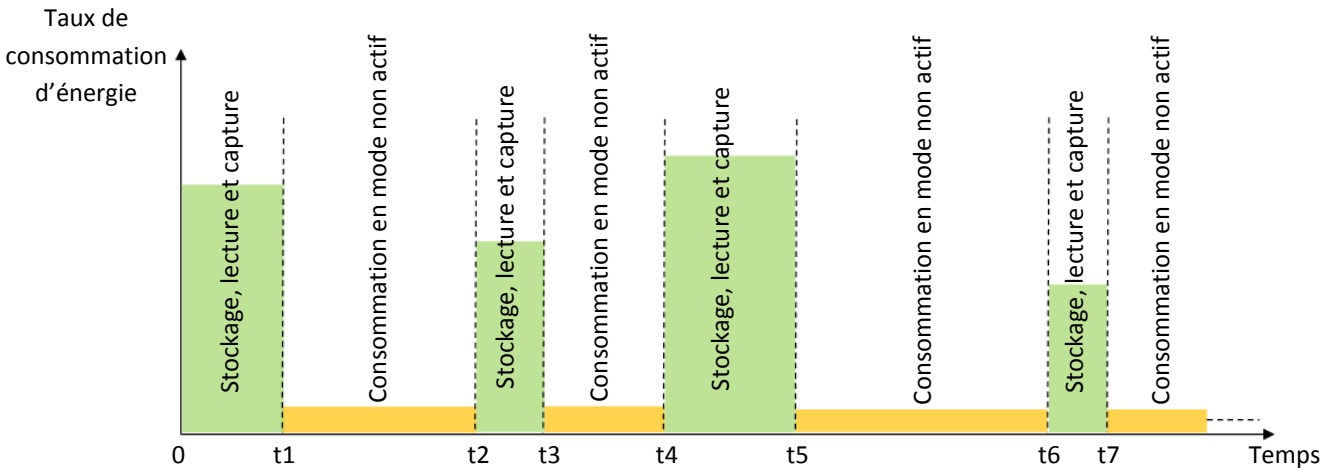


Figure 86. Evolution de la consommation d'énergie des nœuds durant le cycle de vie du préfabriqué

Conclusion générale et perspectives

Cette thèse s'inscrit dans le cadre de la dissémination des informations liées au produit tout au long de son cycle de vie et plus particulièrement dans le contexte de l'habitat. Ces travaux de thèse proposent des protocoles de stockage et de lecture des informations relatives aux préfabriqués en béton en se basant sur le concept de la matière communicante. Ces protocoles visent à stocker puis à lire des informations dans les préfabriqués pour des fins de traçabilité, surveillance, maintenance, recyclage, etc. Pour accomplir cet objectif, des centaines de micro-nœuds sont déployés uniformément dans le béton afin de former un réseau de capteurs dans le préfabriqué.

Le chapitre 1 retrace l'idée du concept de matière communicante comme nouvelle technologie d'Internet des Objets, ainsi que les premiers travaux relatifs à ce concept développés au Centre de Recherche en Automatique de Nancy. Ensuite, une étude sur la gestion du cycle de vie des produits (PLM) a été présentée. Le chapitre donne finalement notre positionnement et notre problématique dans le contexte de construction des préfabriqués en béton. La problématique technique est de rendre le préfabriqué communicant en intégrant des centaines de micro-nœuds dans le béton, tandis que la problématique scientifique est de proposer de nouvelles solutions de stockage et de lecture des informations dans la mémoire des micro-nœuds enfouis dans le béton du préfabriqué.

Dans le chapitre 2, un état de l'art sur les réseaux de capteurs a été présenté en montrant notamment des travaux de recherche proches de notre problématique de façon à sélectionner des solutions existantes potentiellement utilisables pour construire de nouveaux protocoles de communication plus adaptés à notre contexte. Le premier constat est qu'il existe peu de travaux qui cherchent à disséminer de façon uniforme des informations dans les nœuds d'un réseau de capteurs. Le second constat est que le stockage de l'information se fait sans considérer la sémantique de l'information, c'est-à-dire sans considérer le niveau d'importance d'une information par rapport à une autre. Le troisième constat qui découle de la conséquence des deux premiers est qu'il n'existe pas dans la littérature de protocoles spécifiquement conçus pour lire les informations uniformément dispersées dans des nœuds de capteurs.

Face à ces différents constats, nous avons donc spécifié de nouveaux protocoles pour répondre aux points suivants :

- Comment stocker uniformément l'information dans un réseau de capteurs pour que l'information soit présente sur toute la matière communicante ?
- Comment faire varier la densité de stockage dans un réseau de capteurs selon le niveau d'importance de l'information ?

- Comment lire efficacement les informations uniformément stockées dans le réseau de capteurs ? Le terme efficacité se réfère à des contraintes de fiabilité, de délai, de congestion (trafic) et enfin d'énergie (car les batteries des nœuds enfouis dans la matière sont difficilement accessibles et donc rechargeables).

Pour répondre à ces trois questions, nous avons proposé dans la première partie du chapitre 3 un protocole de dissémination des informations, nommé USEE (Uniform Storage and Energy Efficient data dissemination protocol), qui se compose de trois techniques clés :

- L'inondation « counter-based » pour transmettre l'information vers tous les nœuds dans la matière.
- Le stockage probabiliste afin de répliquer l'information selon leur importance dans les différents nœuds de la matière.
- La réplication contrôlée dans chaque voisinage afin de résister à la surcharge des mémoires des nœuds.

Dans la deuxième partie du chapitre 3, nous avons proposé un protocole de lecture d'informations, nommé RaWPG (Random Walk and Pull Gossip data retrieval protocol), adapté au protocole USEE. Le protocole RaWPG est basé sur quatre techniques clés :

- La transmission des messages requête et réponse selon un mécanisme de marche aléatoire « Random Walk » permettant de réduire le taux de duplication des messages et d'optimiser la consommation d'énergie.
- Le mécanisme « Pull Gossip », intégré au niveau de chaque saut du Random Walk afin de couvrir un grand nombre de nœuds dans le processus de recherche.
- Un algorithme de sélection des nœuds de bordure, augmentant à chaque saut le nombre de nouveaux nœuds couverts au prochain saut de Random Walk et ayant pour conséquence d'augmenter la chance de trouver l'information à chaque saut.
- Une fonction de coût de liaison utilisée afin d'acheminer les messages requête et réponse par des chemins fiables à haute qualité de service.

Les techniques de stockage et lecture d'information ont été proposées en tenant compte aussi du contexte de la matière communicante qui se distingue par : une forte densité des nœuds de capteurs enfouis dans le produit béton, une utilisation des micro-nœuds avec des ressources limitées (mémoire, énergie et capacités de calcul), et dont le remplacement des batteries/nœuds est quasiment impossible. Ces contraintes ont imposé une optimisation des paramètres de chaque protocole, et le développement de modèles de communication simples.

Nous avons évalué la performance des protocoles USEE et RaWPG par simulation dans le chapitre 4. L'outil de simulation Castalia/OMNeT++ a été choisi car il offre un environnement de simulation plus proche de la réalité pour les réseaux de capteurs sans fil. Par ces simulations, nous avons montré que les performances des protocoles augmentent quand on utilise un grand nombre de nœuds dans chaque voisinage dans le réseau de capteurs.

Le protocole de stockage montre une uniformité de stockage, un taux de réplication contrôlé ce qui permet une grande capacité de stockage de la matière, et il consomme peu d'énergie par rapport à d'autres protocoles de la littérature. Quant au protocole de lecture, il a montré aussi de bonnes performances en termes de taux de succès pour retrouver l'information et de consommation d'énergie. De plus, aucune duplication de réponses est nécessaire comparativement aux protocoles de la littérature dans les réseaux de capteurs qui sont basés en majorité sur l'inondation de requêtes.

L'étude par simulation des protocoles USEE et RaWPG a aussi été réalisée en tenant compte du contexte applicatif. Les résultats montrent que ces deux protocoles utilisées pour suivre le cycle de vie d'un préfabriqué en béton ou plus spécifiquement d'un bâtiment répondent bien aux exigences des parties prenantes tant sur les phases de lecture que d'écriture.

L'objectif de cette thèse en trois ans était de développer de façon théorique des protocoles d'écriture et de lecture pour la matière communicante. Tous ces résultats théoriques ont été publiés dans quatre conférences internationales et dans deux revues internationales indexées. Les résultats sur le protocole de lecture sont actuellement soumis dans une troisième revue. La suite de nos travaux va donc consister à passer à la phase d'ingénierie en implémentant nos protocoles dans des nœuds de capteurs et en les testant dans des situations applicatives réelles.

Les perspectives sont donc à court terme :

- D'utiliser la plateforme FIT-IOT LAB (www.iot-lab.info) : le programme compilé de chaque protocole peut être envoyé à cette plateforme à distance via un portail web, afin de programmer un ensemble de nœuds installés dans six institutions en France. Cette plateforme permet de tester les algorithmes sur un réseau de densité maximale actuelle de 2728 nœuds, ce qui est proche du contexte de la matière communicante.
- D'intégrer des nœuds de capteurs dans un produit réel en béton (développement d'un prototype) pour analyser les résultats des protocoles vis-à-vis de l'effet des matériaux dans le béton sur la propagation signal radio (perturbation, interférence, affaiblissement, etc.). Comme présenté dans le premier chapitre, le CRAN travaille en collaboration avec le Centre d'Études et de Recherches de l'Industrie du Béton (CERIB) situé en France (www.cerib.com). Notre travail va donc consister à développer avec CERIB des murs intelligents capables de gérer des informations statiques (par ex. origine, matière, dimensions du préfabriqué) et des informations dynamiques (par ex. luminosité, humidité et température dans le béton). Ces murs intelligents permettront donc de tester en grandeur réelle les protocoles USEE et RaWPG pour gérer le cycle de vie du bâtiment.
- De spécifier le format des données qui seront écrites dans la matière par le biais de modèles standards en s'appuyant par exemple sur le format BIM-IFC.

- Et enfin d'étudier les mécanismes de réveil et d'endormissement qui seraient les plus adaptés aux protocoles USEE et RaWPG pour prolonger la durée de vie des nœuds.

À plus long terme, plusieurs perspectives s'offrent à nous :

- Améliorer l'efficacité des protocoles USEE et RaWPG en utilisant la compression de données pour stocker un plus grand nombre d'informations.
- Etendre ces protocoles de manière à ce qu'ils puissent gérer la vie de l'information (effacement progressif si elle n'est pas utilisée), fusionner des informations proches voire même inférer de nouvelles informations.
- Envisager d'autres applications possibles pour les protocoles USEE et RaWPG. On pourrait par exemple imaginer d'utiliser ces protocoles dans l'aéronautique pour disséminer les informations des boîtes noires sur toute la structure de l'avion. En cas d'incident, le fait de retrouver une partie de la carlingue permettrait de récupérer de nombreuses informations sans être tributaire d'une recherche très aléatoire des boîtes noires.

Annexe : Comparaison entre le stockage probabiliste et le stockage par compteur de nombre de sauts

Computer Standards & Interfaces 44 (2016) 243–257

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Non-localized and localized data storage in large-scale communicating materials: Probabilistic and hop-counter approaches

Kais Mekki [a,b,c,*], William Derigent [a,b], Ahmed Zouinkhi [c], Eric Rondeau [a,b], André Thomas [a,b], Mohamed Naceur Abdelkrim [c]

[a] Université de Lorraine, CRAN, UMR 7039, 2, avenue de la forêt de Haye, Vandoeuvre-lès-Nancy Cedex 54516, France
[b] CNRS, CRAN, UMR 7039, France
[c] Research Unit of Modeling, Analysis and Control of Systems (MACS), National Engineering School of Gabes, rue Omar Ibn Elkhattab, 6029 Gabes, Tunisia

ARTICLE INFO

Article history:
Received 15 December 2014
Received in revised form 22 June 2015
Accepted 19 August 2015
Available online 2 September 2015

Keywords:
Internet of Things
Communicating materials
Wireless sensor networks
Data storage
Dissemination protocols

ABSTRACT

The rapid development of Internet of Things has triggered the multiplication of communication nodes based on Radio-Frequency Identification (RFID) and Wireless Sensor Networks (WSNs) in various domains such as building, city, industry, and transport. These communication nodes are attached to a thing or directly included in the material of the thing to form a communicating material. In communicating material, one of the desired objectives is to merge the logical data with its physical material, thus simplifying the monitoring of its life cycle, the maintenance operations, and the recycling process. In this context, the initial form of the communicating material can evolve during its lifecycle. It can be split, aggregated with other materials, or partially damaged. However, the entire information in the material should always be accessible after each change. Thus, the objective of this research is to develop specific algorithms for efficient dissemination of information in the material in order to limit information losses. Two dissemination algorithms hop-counter-based and probabilistic-based are proposed for storing data by using WSNs, and non-localized and localized storage is considered. Non-localized storage ensures that information can be retrieved from each piece of the material by using a uniform data replication process. Localized storage ensures that the information is stored in a limited region of the material. Castalia/OMNeT++ simulator is used to compare the performance of the proposed algorithms with other similar protocols such as DEEP, Supple, and RaWMS.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Communicating material, a new paradigm of industrial information systems, was presented and discussed for the first time in [1]. It enhances a classic material by providing the following capabilities: (a) storage of data, (b) communication of information at any point of its surface, and (c) retention of properties (a) and (b) after physical modifications. This concept leads to an important change in the Internet of Things. During product manufacturing, thousands of ultra-small electronic devices are distributed in its material. Thus, the product does not communicate using certain tags or nodes at specific points, but communicates intrinsically and continuously.

The initial studies focusing on communicating materials are presented in [1–6]. In these studies, a communicating material called *e-textile* is obtained by scattering a large number of Radio Frequency Identification (RFID) µtags (1500 tags/m^2) in a manufactured textile. The system consists of an RFID reader/writer connected to a relational database that contains the entire product lifecycle information. For each writing operation, the database is examined to select the relevant data items (fragments of the database tables) that must be stored in the material. In order to achieve this objective, each data item is assigned an importance level between 0 and 1 that is computed via a multi-criteria decision-making algorithm [5]. For example, the importance value 1 indicates a highly critical data item and the value 0 indicates an ordinary one. Then, data items with the highest importance levels are stored in the µtags when the textile passes under the writer module during the manufacturing phase. The RFIDs are memory-constrained, and hence, the data item is split and stored over several tags by using a specific protocol header which can be used to rebuild the initial information. This division process is called segmentation and the resulting parts are *segments*.[1]

* Corresponding author at: Université de Lorraine, CRAN, UMR 7039, 2, avenue de la forêt de Haye, Vandoeuvre-lès-Nancy Cedex 54516, France.
E-mail addresses: kais.mekki@univ-lorraine.fr (K. Mekki), william.derigent@univ-lorraine.fr (W. Derigent), ahmed.zouinkhi@enig.rnu.tn (A. Zouinkhi), eric.rondeau@univ-lorraine.fr (E. Rondeau), andre.thomas@univ-lorraine.fr (A. Thomas), naceur.abdelkrim@enig.rnu.tn (M.N. Abdelkrim).

[1] In our article, the term *segmentation* is not employed in its classic sense which is typically used for the transport layer of the OSI model.

http://dx.doi.org/10.1016/j.csi.2015.08.010
0920-5489/© 2015 Elsevier B.V. All rights reserved.

In such a system, data storage in the communicating material requires a reader/writer connection with each tag. If a tag is not connected during the dissemination phase, it will be isolated and left empty, thus limiting the use of RFID technology in solid and large-scale materials such as concrete in smart buildings, plane wings, and wood panels. Therefore, this study proposes the use of Wireless Sensor Networks (WSNs) in such products by dispersing micro sensor nodes in the material, as shown in Fig. 1. The insertion of WSNs into materials has been proposed by [7]; however, the objective of the insertion was self-measurement purposes. In our study *focusing on data management*, a dissemination algorithm for WSNs is used to store relevant data items in the material.

During the manufacturing process of a final product and its future use, the communicating material passes through various lifecycle steps, as shown in Fig. 2. In steps 1 and 2, the material undergoes shape transformation (e.g. cutting, sawing, and drilling) to construct the product. In the intermediate stage of the lifecycle (steps 3 and 4), the material could also undergo possible physical transformations (e.g. breakage, piece addition/elimination) which lead to information loss if data is not replicated in the entire material. Therefore, the information should be stored in a uniform manner, and it should be present in each piece of the material. Further, by ensuring uniform distribution, the data item can be retrieved later by the user without visiting many nodes in the material, thus conserving its energy resources.

WSNs in communicating material exhibit certain specific characteristics: high node density and ultra-small nodes (micro-nodes) with extremely limited memory, energy, and computation abilities. Further, replacement of nodes or batteries is impossible because the nodes are embedded in the material. Hence, the data dissemination in such networks and environment must be judicious. The amount of data transmitted in the network should be reduced as much as possible to maximize the material lifetime (i.e. communication is the main activity responsible for energy consumption in WSNs [8–10]). Thus, any data dissemination mechanism developed for nodes with such limited resources must be simple and incur low communication overhead.

Another factor in the storage in communicating materials is the data importance level, as discussed in [5]. In order to disseminate information, the user sends a data item coupled with an importance level having a value between 0 and 1, which affects the storage density (i.e. the data replication rate). The number of storage nodes should depend on this level. For instance, the number of storage nodes for an importance level of 1 should be larger than that for lower levels of importance. Thus, the key research challenge of this study is to determine a mechanism for storage of the data item among limited-resource WSNs; thus, the data item will be present in each piece of the material (uniform storage), irrespective of the importance level. A failure in this process may result in data loss during material transformation in its lifecycle.

Considerable research has been conducted in data dissemination in WSNs [11]; however, few studies have considered the challenges mentioned earlier in this manuscript. In general, dissemination algorithms can be categorized as *reactive* or *proactive*. In the reactive approach [12–15], nodes react to an event by sending data towards the nodes that are located close to the event positions. However, in the proactive approach, nodes anticipate future events and distribute their data towards all the nodes or a subset of nodes that have the role of a storage unit. This proactive approach is composed of *structured* and *unstructured* algorithms. In structured dissemination [16–18], the storage nodes typically form a virtual structure (e.g. grid, line, and rail) within the WSN, which makes the data available for retrieval at a later point of time (e.g. by sinks visiting the storage nodes to collect data). However, in unstructured dissemination [19,20], the data is replicated throughout the entire network. In this case, the manner in which data dissemination is performed will determine whether the information is uniformly stored. *In this study, we focus on unstructured proactive data dissemination strategies and selection of uniform distributed storage nodes in WSNs.*

In this study, we have developed two unstructured proactive dissemination algorithms for the storage of data items in large-scale communicating materials. Both algorithms use a counter-based broadcasting scheme for spreading the message to all nodes of the material. However, the algorithms have different storage strategies. The first algorithm uses probabilistic-based storage and the second algorithm uses hop-counter-based storage.

- *Probabilistic-based storage*: Every node that receives data stores it with a probability P which is equal to the importance level.
- *Hop-counter-based storage*: The message is broadcast from one node to all its neighbours, and, at each hop, the counter is decremented. When the counter is zero, the node must store the data, reset the counter to its initial value, and rebroadcast the message.

As shown in Fig. 3, various storage modes are proposed for these dissemination algorithms: *non-localized storage* and *localized storage*. In the non-localized mode, the data is replicated in a uniform manner, and thus, it will be readable everywhere on the material, even after a shape transformation. The storage density varies in the material according to the data importance level. However, in the localized mode, the storage region size should depend on this importance. The storage region is larger for higher values of importance than for lower values.

These algorithms are simulated with Castalia/OMNeT++ using a realistic collision model and compared with other unstructured proactive data dissemination schemes such as DEEP, Supple, and RaWMS. They are evaluated by studying the uniformity performance for various importance levels, the communication overhead ratio, and the efficiency of the localized storage.

The remainder of the manuscript is organized as follows: In Section 2, related work is discussed. The design of the dissemination algorithms for data storage in communicating materials is described in Section 3. Simulation results and performance evaluation are discussed in Section 4. Finally, the conclusion is presented in Section 5.

2. Related work

Unstructured proactive dissemination protocols have been proposed in literature in order to overcome the challenge of node failures and mobile collector sink management. The goal of replication is to copy data at other nodes within the WSN in order to increase resilience and improve

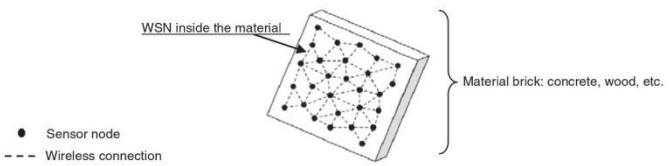


Fig. 1. Large-scale communicating material using WSN.

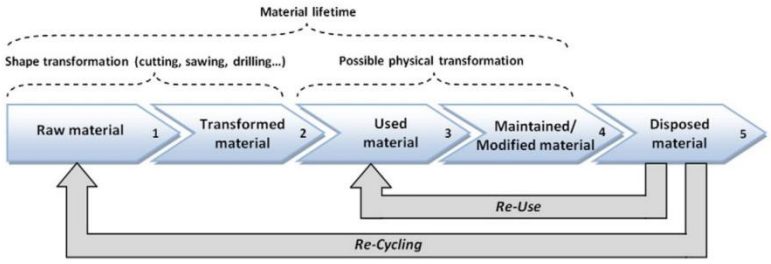


Fig. 2. Communicating material lifecycle.

mobile sink information gathering without the necessity of visiting all nodes.

Unstructured proactive dissemination protocols in WSNs include *broadcast mechanism* and *storage strategy*, which are described below:

2.1. Broadcast mechanisms

Broadcast algorithms are generally referred to as flooding. Flooding is an important algorithm in WSNs, and is applied when a source node must send information to all nodes or a subset of nodes in the network. It is achieved by broadcasting a message to the entire neighbourhood. Each node that receives the message rebroadcasts the message if it has not been forwarded earlier. Thus, the information traverses the entire network and reaches all the nodes which then decide whether to store it based on the storage strategy. Although flooding is an extremely simple and efficient mechanism for data dissemination, it has certain limitations. The main challenges are: duplication (i.e. a node is forced to receive information twice from two different nodes), collision (i.e. the broadcast increases contention), and resource blindness (i.e. nodes do not adopt energy-saving mechanisms) [21]. As a consequence, various schemes, including probabilistic-based schemes, have been proposed for controlled flooding. These schemes are also referred to as gossiping [22,23], counter-based [24,25], distance-based, and location-based [26] schemes.

Distance-based and location-based schemes utilize information related to the distance between nodes and their position. Therefore, nodes must be equipped with a Received Signal Strength Indicator (RSSI) or a Global Positioning System (GPS). Such additional functions could not be applied in communicating material owing to the high density with which the nodes are embedded in the product.

In [27,28], the authors show that the counter-based scheme outperforms the probabilistic one with respect to reliability (i.e. the average percentage of nodes in WSN that are reached) and efficiency (i.e. the average amount of resources required for broadcasting a message in the entire network). The counter-based mechanism can reduce the number of retransmitting nodes having a high arrival rate. Further, it does not require specific hardware such as that required in distance-based and location-based schemes. Hence, the counter-based strategy is adopted as the broadcast mechanism in our dissemination algorithms.

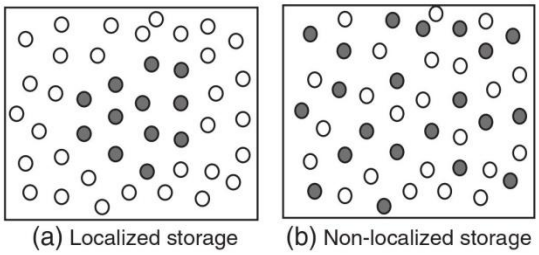


Fig. 3. Storage modes for data item dissemination in the material.

2.2. Storage strategies

In literature, storage strategies in unstructured proactive dissemination protocols have been developed based on the target application. In [19,20], the information is replicated in each node for network reprogramming. If a node does not receive the new version of binary code, its software is not updated and is isolated. Authors in [29,30] propose storage strategies to improve network resilience against the risk of node failures. The storage nodes are selected based on some critical parameters such as connectivity, available memory, and remaining energy. DEEP [31] adopts another storage strategy for effective data collection by a mobile sink with uncontrolled trajectory. Every node that receives data stores it with a probability *ps* based on the amount of information that must be replicated in each node. Supple [32] is another probabilistic approach which uses the tree topology. The root node is responsible for receiving and disseminating the data in the network. This node transmits the data a certain number of times using the tree infrastructure, and each node *i* stores it based on its storage probability $W(i)$. In RaWMS [33], the authors use a random walk mechanism to replicate data for mobile sink collection. This approach is based on a reverse maximum degree random walk (*RW*) sampling technique. Each source node starts an *RW* many times until the data is replicated in the entire network. Each *RW* traverses the network for a predefined number of hops; the last node is responsible for storing the data.

2.3. Synthesis and proposed solution

The main broadcast and storage approaches for unstructured proactive data dissemination algorithms in WSNs can be classified into two categories, as shown in Table 1.

The existing storage strategies use the physical node parameters (e.g. memory and energy), network topology, and neighbourhood awareness to replicate data throughout the WSN. They do not assume the properties and the characteristics of the disseminated data. This limitation is the key focus of our study in which dynamic information-based dissemination algorithms are proposed and applied to the communicating material.

In this study, the data importance level is processed. The item with a high level of importance must be replicated to a greater extent than that with a lower level of importance. The dissemination algorithm must ensure the uniformity of replication for each level.

Table 1
Broadcast and storage approaches.

	Broadcast mechanism	Storage strategy
Probabilistic approach	Probabilistic-based [21,22]	Probabilistic-storage [30,31]
Deterministic approach	Counter-based [23,24]	Hop-counter [32]

In literature, few studies attempt to compare data dissemination protocols in the entire WSN, and, from our knowledge, replication uniformity has not been studied. Hence, probabilistic and hop-counter approaches coupled with counter-based flooding are developed in this study, and the results are compared from a uniformity perspective. The remainder of the manuscript describes the proposed data dissemination algorithms in detail, and then, presents simulation results obtained for various data importance levels.

3. Proposed data dissemination algorithms

This section presents our strategies and algorithms for data item dissemination in communicating material. In this section, first, the concept of master node and the node operations for data dissemination are defined. Then, the counter-based forwarding scheme is described, and finally, the storage strategies are presented.

3.1. Master node

In order to disseminate information, the user chooses a node in his transmission range to connect to the communicating material, as shown in Fig. 4. In this manuscript, this chosen node is called "master node". *Any node in the material could be a master; it depends on the choice of the user during the process of connection to the material.* Certain constraints such as the highest residual energy level could be used in the selection of the master node. The master initiates the data dissemination towards the rest of the network: after receiving the message from the user, the master begins to broadcast the message towards its neighbour nodes.

3.2. Node states for data dissemination

Each node that executes the proposed algorithms assumes different states during data dissemination, as shown in Fig. 5:

① Idle state: The node is non-active and its radio module is in standby mode to conserve its energy.
② Reception mode: The node receives the message from one of its neighbours in the WSN. The received message contains the data and its associated importance level.
③ Storage decision: The node decides whether to store the data by using a function that takes the importance level as parameter. The storage decision also depends on the available memory in the node.
④ Broadcast mode: The node broadcasts the message within its neighbourhood to continue the data dissemination towards the neighbours that have not received this message earlier.

The switch from one state to another is described in the UML sequence diagram depicted in Fig. 6. When a message is broadcast towards the node, the node state changes from the idle state to the reception mode. In the reception mode, the node receives the data and its associated importance level. Then, it commences the storage decision. Based on the result of the storage function, the node

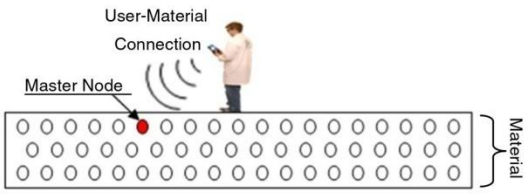


Fig. 4. Master node selection during the user-material connection.

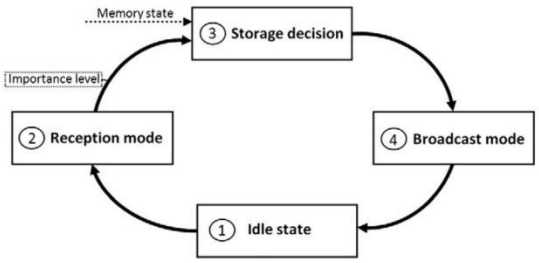


Fig. 5. Node states during data dissemination.

determines whether to store the information. The node memory state is also considered; in the absence of sufficient memory space, the data storage is cancelled. Irrespective of the outcome of the storage decision, the node enters the broadcast mode, in which it transmits the message to its neighbours. In the broadcast mode, the dissemination process continues until the message reaches all the nodes in the material. In the last step, the node returns to the idle state.

3.3. Broadcast strategy: counter-based scheme

Our dissemination algorithms adopt the counter-based broadcasting scheme to transmit the message efficiently through the entire material. A detailed description of this scheme is provided below.

In simple flooding, as the number of duplicate broadcasted messages received by a node increases, the effectiveness of its rebroadcasting decreases [27]. The reason for this behaviour is that duplicate messages are likely to have been received by the neighbouring nodes. This concept is used in counter-based flooding.

In this scheme, when the number of times that a node has received a redundant message is greater than a predefined threshold *Cth*, the node cancels the rebroadcasting. The counter-based algorithm is described below:

1. When a node receives a broadcasted message for the first time, the node initializes a counter N to 1, and sets a random assessment delay (RAD) to a value between 0 and T_{max}.
2. If the node receives the same message during the *RAD*, it increments the counter N ($N \leftarrow N + 1$). If the counter reaches a pre-set threshold *Cth*, the node cancels the rebroadcasting.
3. After the *RAD* expires, the node rebroadcasts the message to its neighbours.

Fig. 7 summarizes the algorithm of this scheme for each node.

Several studies investigate counter-based flooding in order to optimize its parameters, *Cth* and T_{max}. Various methods such as those described in [33,34] could be used for selection of the threshold *Cth*. In [34], a node sets the value of *Cth* based on the number of its neighbour nodes. Similarly, in [35], a node sets the value of *Cth* based on the distance between the node and the broadcasting node. In [36], the authors demonstrate that a threshold value (Cth) between 4 and 6 is preferable from the perspective of the trade-off between reliability and efficiency. Further, simulation results in [28,37] show that if the threshold is 4 or greater than 4, the reliability is greater than 99.5%. Owing to the existence of high-density embedded sensor nodes in the communicating material, the optimal values of the counter-based parameters for communicating material applications was studied in [38]. These values were set as $Cth = 4$ and $T_{max} = 200$ ms based on the trade-off between high reliability and number of retransmitting nodes. These parameter values are consistent with previously optimized interval values for this broadcasting scheme.

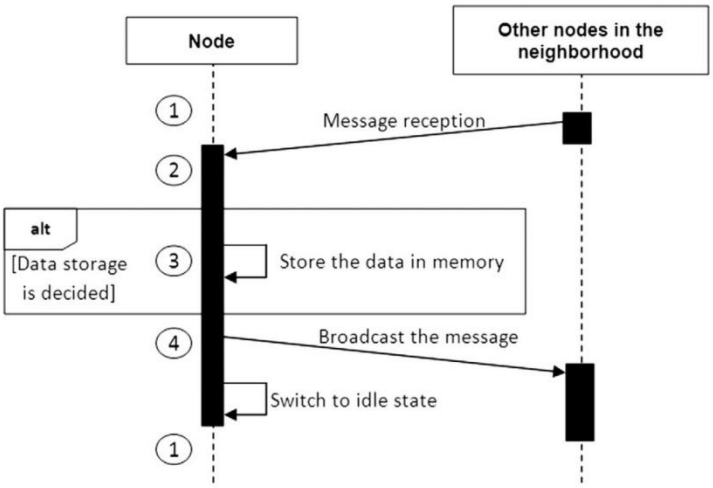


Fig. 6. UML sequence diagram for data dissemination in each node.

3.4. Storage strategy

This section presents two approaches for data item replication in the entire material that use the importance level as a parameter. These approaches are *hop-counter-based* and *probabilistic-based*.

Each data item is coupled with an importance level I. Thus, one aspect of the storage strategy consists of converting this level to a dissemination level that is computed once by the master node and sent to the network of the material. Thus, for each approach, a conversion function F is defined, as shown in Table 2.

In the case of a localized storage, the storage zone must be reduced to an area located around the master node. Thus, for each approach, a limitation function G is defined, which adapts the dissemination level (previously computed by the master node using F) to the storage zone. This function is executed on each node that receives the data item (Table 2).

3.4.1. Hop-counter-based storage

The master node begins to broadcast the message by using a hop counter value equal to H_0, which is computed with the conversion function F_1 and the importance level I (Table 2). F_1 provides the number of hops H_0 for each interval subset of I, as shown in Eq. (1). For example, $F_1(I) = H_0 = 1$ if $0.85 < I \leq 0.9$.

$$\forall I \in [0,1], H_0 = \begin{cases} 0 & \text{if} \quad 0.9 < I \leq 1 \\ 1 & \text{if} \quad 0.85 < I \leq 0.9 \\ 2 & \text{if} \quad 0.8 < I \leq 0.85 \\ & \vdots \\ 20 & \text{if} \quad 0 < I \leq 0.1. \end{cases} \tag{1}$$

The message is broadcast from one node to all its neighbours, and at each hop, the counter H_0 is decremented ($H_0 \leftarrow H_0 - 1$) until it reaches zero. When the value is zero, the node stores the data, resets H_0 to its initial value, and rebroadcasts the message within its neighbourhood. This process continues until the edges of the material are reached.

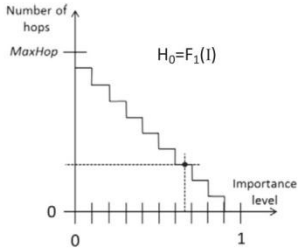
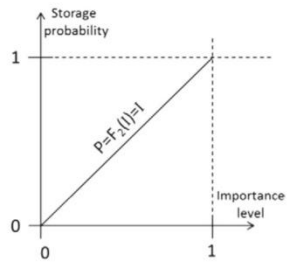
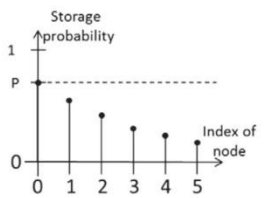
Our dissemination algorithm is broadcast-based. In this communication model, the message is further processed by the nodes that have already received it. Further, the user can disseminate many segments or other data items. Therefore, each message is identified in the network layer (Message*ID*), and then, the nodes are restricted to accept the same message only once. If a node receives a message with the same ID again,

Algorithm: Counter-Based Broadcasting Scheme

```
For a node X
On reception of a broadcasted message m for the first time
    - Initialize the counter N to 1
    - Set RAD and wait for it to expire
    - While waiting:
            For every duplicate message m received, increment N by 1
    - When the delay RAD expires
        - If (N<Cth)
            Rebroadcast the message m
        - Else
            Drop the message m
End
```

Fig. 7. Algorithm of the counter-based broadcasting scheme.

Table 2
Storage approaches.

	Hop-counter-based	Probabilistic-based
Dissemination level	H_0: number of hops between two strorage nodes.	P: probability of storing the data item.
Conversion function F for non-localized dissemination (executed only on the master node)	F_1: $[0,1] \rightarrow \mathbb{R}$ $I \rightarrow H_0$  where: - I: importance level of the data item. - *MaxHop:* the maximal number of hops needed to reach all the nodes of the material.	F_2: $[0,1] \rightarrow [0,1]$ $I \rightarrow P=I$  where: - I: importance level of the data item.
Limitation function G for localized dissemination (executed on each node)	G_1: $\mathbb{R} \rightarrow \mathbb{R}$ $H_0 \rightarrow G_1(H_0)$ *if* $NR<S_{NR}$, $G_1(H_0)= H_0$ *else* *stop dissemination* where: - NR: number of replications (initialized to 0); incremented each time a same data message is replicated at a node. - S_{NR}: threshold for the maximum number of replications of a same data message. Computed at the start, it is related to H_0 and R (R: the radius of the storage zone provided by the user): $S_{NR} = \frac{R}{H_0}$	G_2: $\mathbb{R} \rightarrow \mathbb{R}$ $P \rightarrow G_2(P)$ *if* $P>0$, $G_2(P)= \frac{P}{K}$ *else* *stop dissemination* where: - K: a limitation constant, provided by the user. This constant is related to the storage zone. The higher its value, the smaller is the storage zone.  Retransmission of a same data message from node 0 to node 5.

it will drop the message. Fig. 8 shows the flowchart of this process executed at each node.

3.4.2. Probabilistic-based storage

In this solution, the messages are flooded through the entire material and they reach all the sensor nodes by the counter-based flooding process. Each node decides to store data locally with the probability P that is determined by the conversion function F_2. As an example, let $P = I = 0.5$. On receiving a message, a node selects a random value *rand* within the interval [0,1]. The node stores the data item only if $rand \leq P(P = 0.5)$. Fig. 9 shows the flowchart that describes this process executed at each node. The counter-based flooding process is similar to that shown in Fig. 8, and hence, only the storage process is illustrated in Fig. 9.

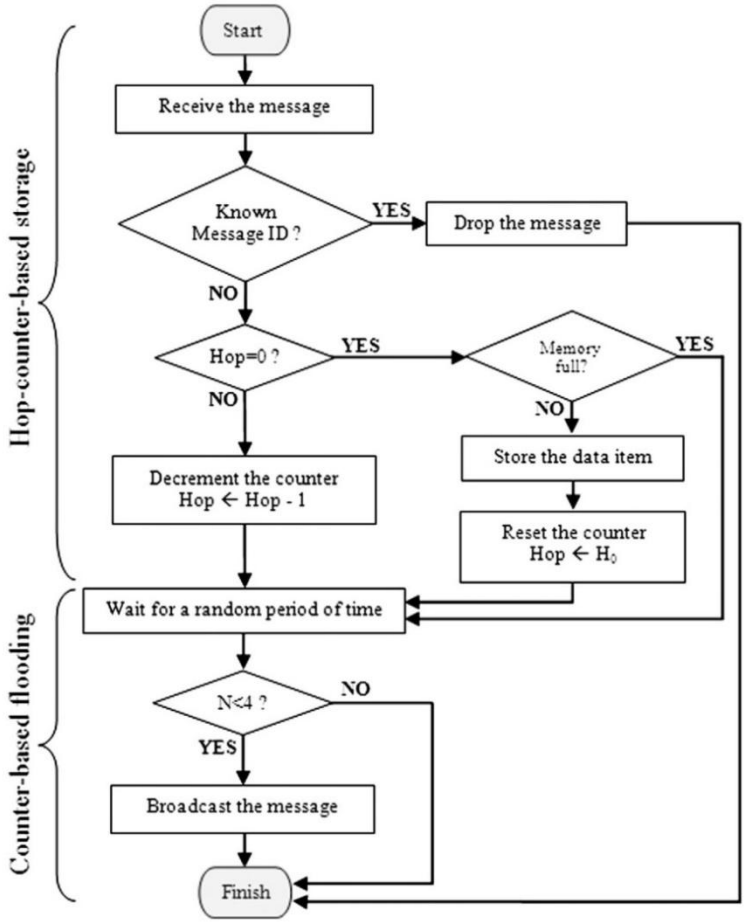


Fig. 8. Flowchart of hop-counter data dissemination algorithm.

3.5. Localized storage strategy

The two proposed algorithms can be completed with limitation functions that are defined in order to restrict the storage zone around the master node. For the hop-counter-based approach, G_1 is defined as a piece-wise linear function that influences the value of H_0 (Table 2). Fig. 10 represents the effect of the limitation function on H_0. First, the message is sent from the master node to the network. When the number of replications *NR* (number of times that the data item has been replicated) is higher than a threshold S_{NR}, the dissemination process is stopped.

For the probability-based approach, the limitation function G_2 is the division of the initial probability *P* by a constant *K*, as shown in Table 2. As we pass from one node to another, the probability *P* decreases, moving closer to zero, which prevents nodes from storing the data item, and hence, results in termination of the dissemination process.

Figs. 11 and 12 show the flowcharts describing the localized dissemination algorithm for the hop-counter and the probabilistic approaches, respectively. The counter-based flooding that is used has already been depicted earlier in Fig. 8, and hence, only the storage mechanism is illustrated.

4. Performance analysis

In this section, we evaluate and discuss the performance of our algorithms through computer simulation. First, the simulation settings are described in detail, and then, the performance results are presented and discussed. Two performance metrics are used to evaluate our algorithms:

- Uniformity of data replication: The existence of the disseminated data in each piece of the material. This metric is studied as a function of the data importance level.
- Communication overhead: The number of message transmissions in the material during the dissemination process. This metric is a strong indicator of the energy consumption for sensor nodes because communication is the main activity responsible for energy consumption in WSNs [8–10].

4.1. Simulation environment

The proposed algorithms were implemented with Castalia/OMNeT++ Tools which is an event-driven simulation framework

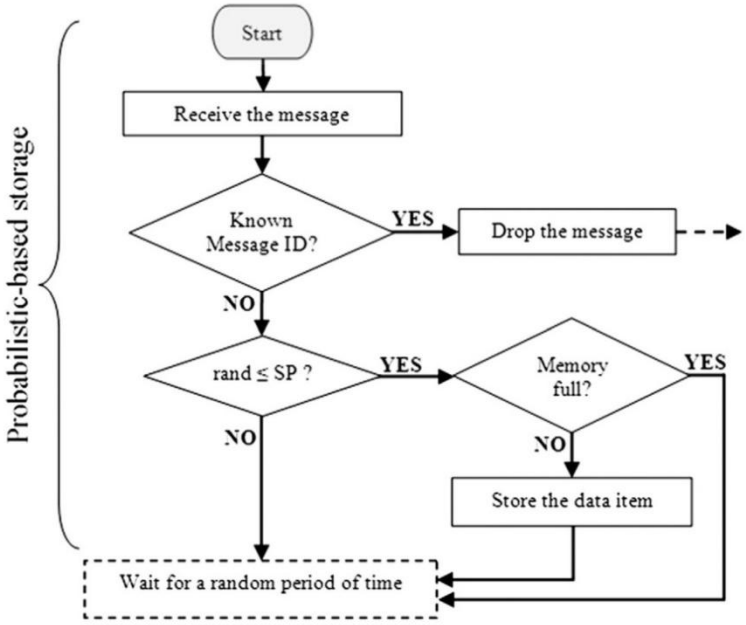


Fig. 9. Flowchart of probabilistic data dissemination algorithm.

written in C++ and widely used in the WSN community. Although many WSN simulators such as COOJA and TOSSIM are available, Castalia provides the most realistic wireless channel, radio models, and sensor node behaviour [39,40].

Castalia is used by researchers to evaluate the characteristics of different platforms for specific applications because it is highly parametric and can simulate a wide range of platforms. In this study, the Tyndall 10 mm sensor node [41,42] is simulated. It is one of the smallest nodes used in the industry and in research (10 mm × 10 mm). Thus, thousands of scattered nodes in a communicating material can be simulated. Table 3 and Fig. 13 show the characteristics of Tyndall 10 mm node.

The simulated communicating material consists of 2500 nodes. All the node positions are uniformly distributed within a 50 m × 50 m square (1 node/m^2). All the nodes are stationary. T-MAC [43], a contention-based medium access control protocol is used as the MAC protocol. Wireless radio channel characteristics such as signal noise, interference ratio, and average path loss are chosen to simulate the realistic modelled radio wireless channel in Castalia based

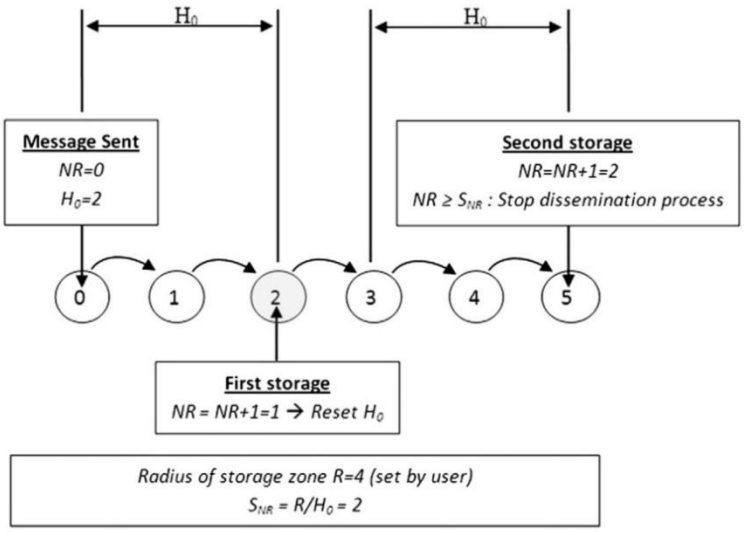


Fig. 10. Localized hop-counter dissemination process.

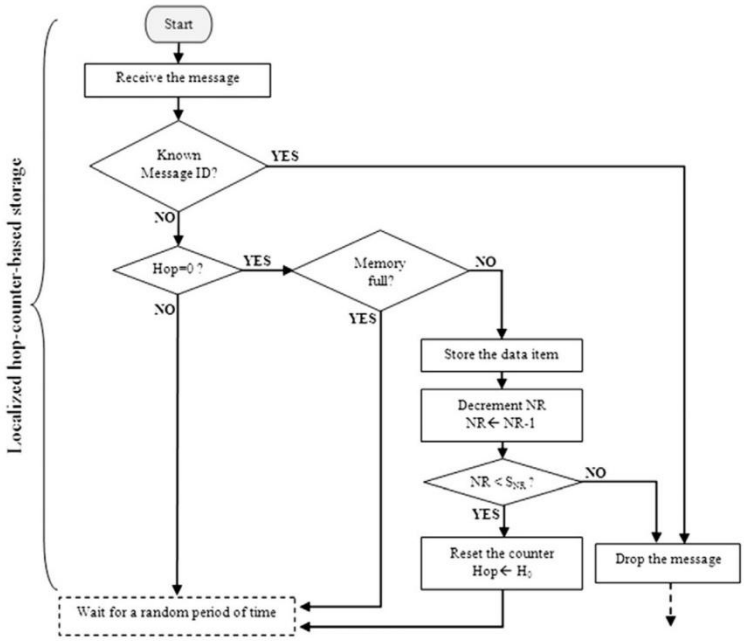


Fig. 11. Flowchart of localized hop-counter dissemination algorithm.

on lognormal shadowing and the additive interference models. A segmented data item is also simulated in this study because a micro-node with limited memory space is used for communicating materials. Each segment is disseminated by a message and the maximum size of the message is fixed as 128 bytes. The parameters used in this simulation study are shown in Table 4.

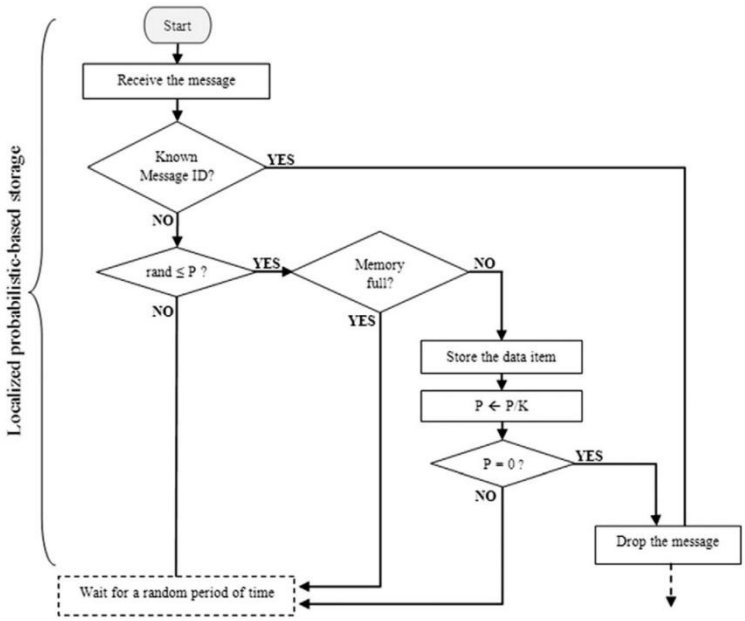


Fig. 12. Flowchart of localized probabilistic dissemination algorithm.

Table 3
Tyndall 10 mm node characteristics.

Size	10 mm × 10 mm	
Microcontroller	Atmega128	Single chip
Transceiver	Nordic nRF905	transceiver/microcontroller
Frequency bands	433 MHz, 868 MHz, and 915 MHz	
Bandwidth	50 kbps	
Programme + data memory	4 kb	
Microcontroller unit	688 μA	
Transmission mode	10.35 mA	
Reception mode	13.32 mA	
Moderate sleep	166 μA	
Deep sleep	3.3 μA	

Table 4
Simulation parameters.

Material size	50 m × 50 m	
Number of nodes	2500	
Node distribution	Grid 50 × 50	
Node density	1 node/m^2	
Mac protocol	T-MAC	
Sigma channel parameter	4	Real radio wireless
Bidirectional Sigma channel parameter	1	channel in Castalia
Radio collision model	2 (additive interference model)	
Cth	4	
T_{max}	200 ms	
Number of data segments	3	
Message size	128 bytes	
Simulation time	100 s	

4.2. Simulation results of non-localized dissemination

4.2.1. Storage and distribution performance

In order to evaluate the storage and distribution performance of our proposed algorithms, first, a non-segmented data item (i.e. one message dissemination) is disseminated in the simulated WSN. Figs. 14 and 15 show the simulation results of the broadcast and storage mechanisms of the hop-counter and probabilistic data replications for various values of hop H_0 and various values of probability P, respectively.

From the results obtained, the following observations can be made: Figs. 14 and 15 demonstrate the high reliability of the counter-based flooding scheme because all nodes store the information for $H_0 = 0$ and for $P = 1$. Further, for the hop-counter and probabilistic storage strategies, the data is replicated in the communicating material based on the importance level. When the importance is low, for example, $I = 0.4$, the data item is stored in 1000 or fewer nodes in the communicating material. The density of storage decreases with a decrease in the importance level for the two algorithms, as shown in Fig. 15. However, the probabilistic algorithm has a higher replication rate than the hop-counter algorithm.

In order to study the distribution performance, the simulated material is divided into small cells of 50 cm × 50 cm, as shown in Fig. 16. Each piece has 25 sensor nodes because a uniform 50 × 50 grid distribution is used. In each cell, the number of nodes that have stored the data item is counted.

Fig. 17 shows the storage density difference between cells, which is obtained by using the standard deviation tool. The standard deviation represents the variable distribution around the mean, and hence, the figure shows that the probabilistic algorithm has a better resulting distribution than the hop-counter approach. When low importance is used (≤0.5), the standard deviation decreases for the hop-counter algorithm (i.e. the distribution is worst with a large value of counter H_0). In this case, many pieces are completely empty and others are poorly filled. However, there is no significant decrease in standard deviation for the probabilistic-based strategy because the data item is uniformly replicated and many pieces are filled for all the importance levels.

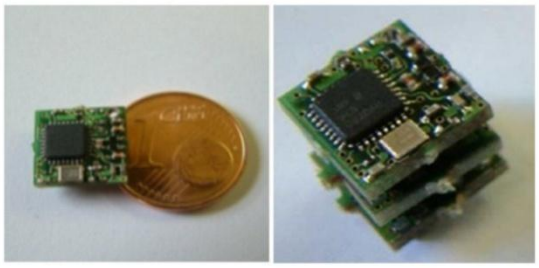


Fig. 13. Tyndall 10 mm node [41].

Therefore, the hop-counter strategy produces more empty areas in the communicating material than the probabilistic-based strategy for low importance levels, which could lead to data loss during material transformation.

4.2.2. Proposed algorithms vs. other solutions in literature

In this sub-section, the performance of segmented data item dissemination is evaluated and compared with the performance of other existing algorithms in literature with respect to the uniformity and communication overhead.

Fig. 18 shows the simulation results for a segmented data item dissemination with the hop-counter and probabilistic strategies. Three segments are disseminated with the same hop-counter value (H_0) for the first algorithm and the same probability P for the second one. The colours black, green, and blue represent the nodes that have stored the first, second, and third segments, respectively.

4.2.2.1. Uniformity performance. In order to study the uniformity of replication and compare the performance of the proposed algorithms with that of DEEP, Supple, and RaWMS, the division of the simulated material into small cells that is shown in Fig. 16 is used. The storage parameters of the other algorithms in literature are varied for each importance level I although they were not designed to support this type of application. $W(inn)$ for Supple and ps for DEEP are equal to I. For RaWMS, the master node starts RW r times based on the importance I. A simple function is adopted; it provides the r value for each interval subset of I, as shown in Eq. (2) (e.g. $r = 2300$ if $0.85 < I \leq 0.9$).

$$\forall I \in [0,1], r = \begin{cases} 2400 & \text{if} \quad 0.9 < I \leq 1 \\ 2300 & \text{if} \quad 0.85 < I \leq 0.9 \\ 2200 & \text{if} \quad 0.8 < I \leq 0.85 \\ & \vdots \\ 100 & \text{if} \quad 0 < I \leq 0.1 \end{cases} \tag{2}$$

This variation has no negative effect on the existing protocols that are evaluated. On the contrary, it leads to a fair comparison with our algorithms for uniformity analysis and communication overhead.

This experiment consists of determining the number of cells of the material that contain the segmented data item. The information exists in a cell if the three segments are present. This experiment provides us information about the extent to which the dissemination algorithms distribute the information over the entire communicating material. The results of this experiment are shown in Figs. 19 and 20. Fig. 19 presents the existence ratio of the information in the cells for the evaluated algorithms (the existence ratio is the number of cells that contain the segmented data divided by the total number of cells). Fig. 20 shows the storage density of the information in the entire network. This storage density is an indicator of the memory overhead.

Fig. 19 shows that the resulting dissemination of our probabilistic-based algorithm and DEEP is more uniform than that of other

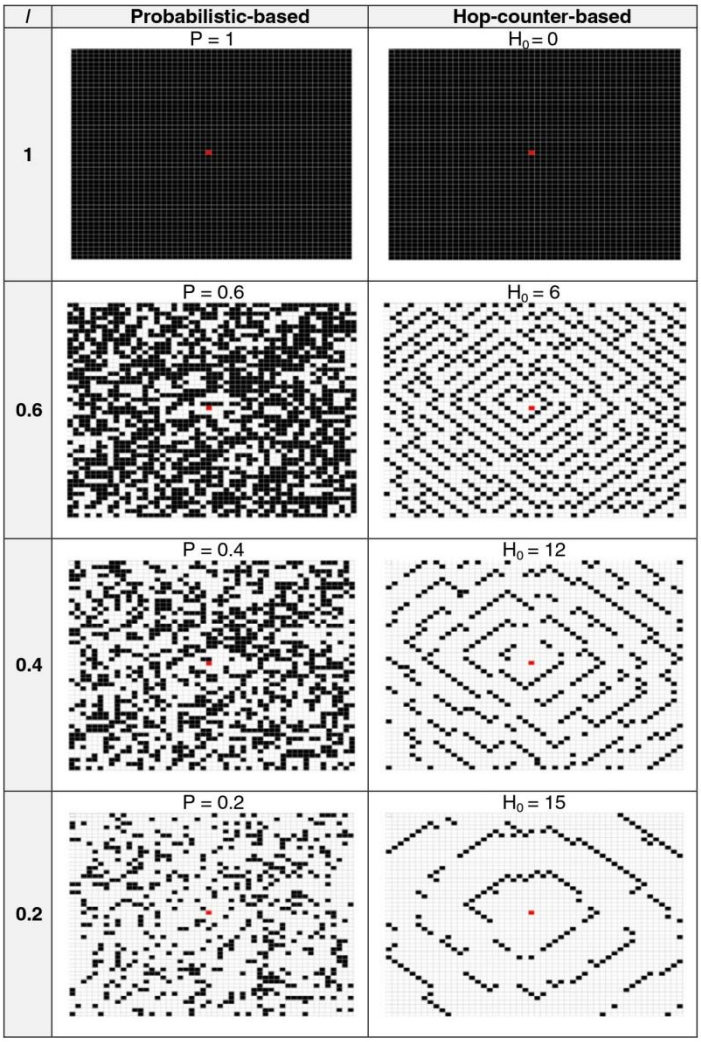


Fig. 14. Non-localized probabilistic storage of non-segmented data item.

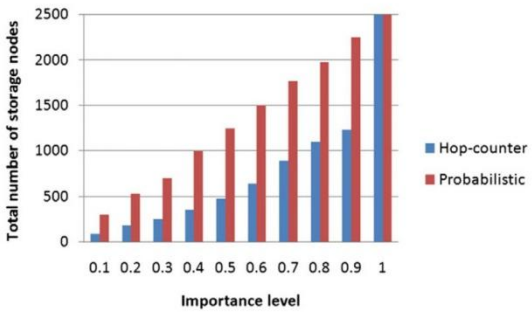


Fig. 15. Total number of storage nodes for various importance levels.

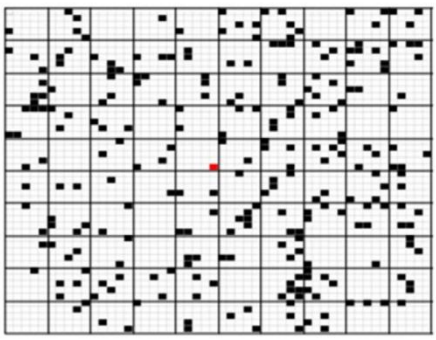


Fig. 16. Division of the material into pieces of 50 cm × 50 cm.

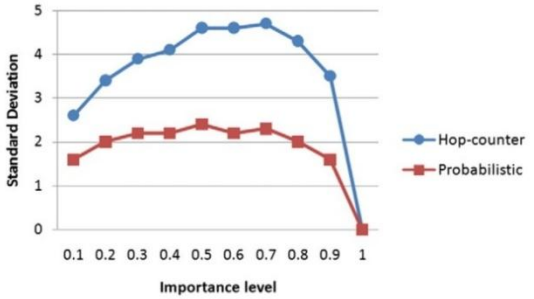


Fig. 17. Importance level versus standard deviation of storage strategies.

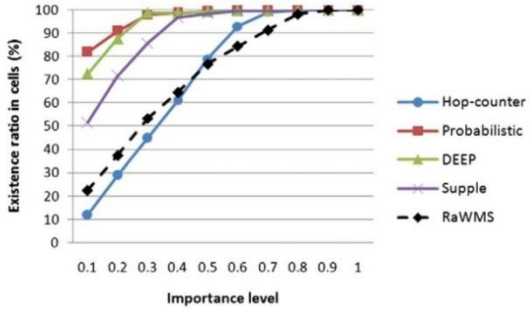


Fig. 19. Uniformity performance of the proposed algorithms and other existing protocols.

algorithms. In the case of our probabilistic-based algorithm and DEEP, for importance levels greater than or equal to 0.3, the information exists in all cells of the material. For these two algorithms, the number of cells in which data is distributed is equal and corresponds to a good data distribution (i.e. greater than 71% and 83% of cells for $I = 0.1$ and $I = 0.2$, respectively, and 100% for all other importance levels). In the case of Supple, the data distribution is 100% for $I \geq 0.4$, and the uniformity decreases for lower importance values (0.3, 0.2, and 0.1), thus yielding more empty data cells.

In the case of our hop-counter-based algorithm, the segments are replicated depending on the H_0 value. When a high H_0 value is used, the segments are stored in a small number of nodes. For instance, when low importance is used (0.4 or less), the segmented data exists in fewer than 60% of the cells (i.e. the uniformity is worst when the H_0 value is high). In this case, many pieces are completely empty and others are poorly filled. However, the information exists in 90% of the cells for $I \geq 0.6$, and a small value of H_0 is required to attain this uniformity. RaWMS shows similar performance; it attains high uniformity for $I \geq 0.8$. Thus, the counter-based replication strategy does not provide a good solution to our problem because it produces many empty areas in the communicating material for low importance values, which could lead to data loss during material transformation. However, the probabilistic-based strategy ensures that a data item is uniformly replicated and many cells are filled for many importance levels; thus, the probabilistic-based storage seems to be the best solution for our problem.

Fig. 20 shows that our probabilistic-based algorithm and DEEP have the highest replication rate (i.e. highest memory overhead) compared to the other algorithms. For instance, the density values are 415, 1112, and 2500 (all nodes) for $I = 0.1$, $I = 0.5$, and $I = 1$, respectively. The data segments are replicated in a large number of nodes, especially for high importance levels, thus resulting in saturation of the memory of the material nodes. Hence, nodes will be unable to respond to any subsequent dissemination. These problems could be resolved by relying on the application of data compression techniques such as those described in [44,45]. Compression algorithms are suited for the reduced storage and limited resources in ultra-small nodes. For example, the compression ratio on environmental datasets can reach 70% [44].

4.2.2.2. Communication overhead. Fig. 21 shows the total number of message transmissions in the network for all evaluated algorithms and for each importance level. It can be observed that the proposed solutions and Supple are the algorithms with the lowest overhead for an importance level greater than 0.6. For instance, our solutions transmit approximately 10% fewer messages than DEEP and 14% fewer messages than RaWMS.

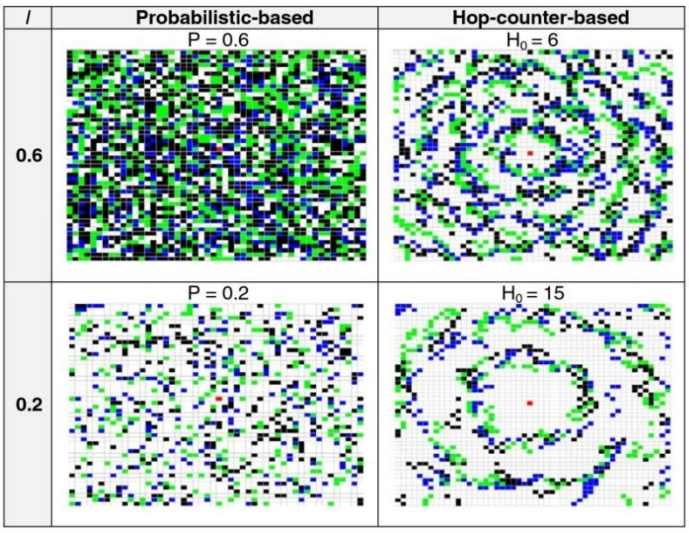


Fig. 18. Non-localized probabilistic storage of segmented data item.

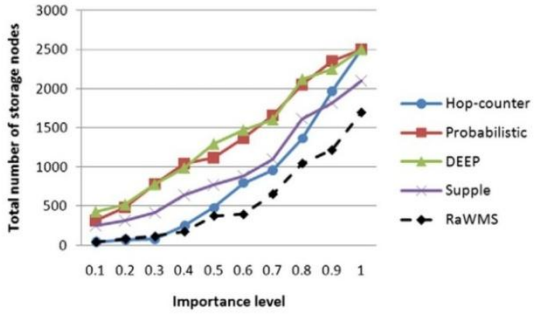


Fig. 20. Total number of storage nodes of the proposed algorithms and other existing protocols.

When an importance less than 0.6 is used, RaWMS has the lowest overhead because it executes the hop-counter process a few times, thus resulting in low uniformity, as described in this section.

Further, our proposed algorithms have less communication overhead than DEEP although they use the flooding scheme to transmit the message in the entire network. The reason for this result is that we used counter-based flooding which generates fewer broadcasting messages than probabilistic flooding used by DEEP, and still ensures high reliability.

Assuming that communication is the main activity responsible for energy consumption in WSNs [10], these results are a strong indication that our solutions will result in low energy consumption for sensor nodes and for all the data importance levels.

4.2.2.3. Performance synthesis. An overview and comparison of the above results is summarized in Table 5, where the storage and transmission strategies are provided with the corresponding uniformity, replication rate, and communication overhead performances. It is evident that the probabilistic storage strategy provides the best uniform distribution. Using this strategy, we can replicate the information in each piece of the material. However, it yields the highest replication rate, which could result in saturation of the memory of nodes.

On the other hand, the hop-counter storage exhibits bad uniformity performance. Many pieces of the material could be empty, which could result in data loss during physical transformations, although it has the lowest and best replication rate. It is evident that a low replication rate leads to bad uniformity performance. The data should be replicated many times in the WSN in order to attain the best uniformity, as observed in the probabilistic strategy.

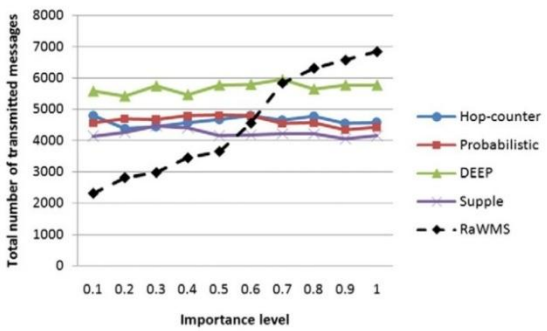


Fig. 21. Communication overhead performance of the proposed algorithms and other existing protocols.

In case of the communication overhead performance, the counter-based flooding and the tree transmission mechanism provide the best and the lowest communication overhead. These broadcast mechanisms reduce the number of transmissions in each neighbourhood, and hence, optimize the energy consumption for data dissemination.

4.3. Simulation results for localized dissemination

4.3.1. Non-segmented data item

Simulation results for localized data item dissemination are shown in Fig. 22 for the hop-counter ($R = 16$) and probabilistic approaches ($K = 2.5$).

From Fig. 22, it can be concluded that the hop-by-hop replication mechanism guarantees efficient data item storage in the target area of the material. The storage density increases based on the hop counter H_0. For example, for $H_0 = 6$, the data item is replicated in 195 nodes. However, for $H_0 = 15$, it is stored in 122 nodes.

In case of the probabilistic approach, the data is also efficiently replicated in the desired area. The density of storage is high in nodes situated around the master; the probability P is high in these nodes and it decreases in nodes located further from the master.

4.3.2. Segmented data item

Simulation results for localized segmented data item dissemination are shown in Fig. 23 for the hop-counter and probabilistic approaches. The colours black, green, and blue represent the nodes that have stored the first (S1), second (S2), and third (S3) segments, respectively.

Initially, the master node disseminates the first segment. If the node memory is full, the node retransmits the message without decrementing the counter. Hence, in the hop-counter algorithm, the second segment is replicated outside the first one and similar behaviour is observed for the third segment. The last broadcasted item is replicated in a larger number of nodes than the other items. For example, for $H_0 = 15$, the first, second, and third segments are stored in 125, 179, and 267 nodes, respectively.

However, in the probabilistic algorithm, the segments are stored in the target area with the same replication rate. All the segments have the same density. Therefore, for localized dissemination, the probabilistic approach is more efficient than the hop-counter approach.

5. Conclusion

In this study, we have investigated methods to disseminate data in communicating material using a wireless sensor network. We focus on the uniformity of dissemination because the material could undergo physical transformations during its entire lifecycle. Further, in this study, we take the data importance level into consideration: highly important data should be replicated in a greater number of nodes than other data. Hence, data should be uniformly disseminated and should be present in each piece of the material, irrespective of the importance level. In addition, we focused on storing the data in a limited region of the material. The region size should be fixed based on the importance level; the region size is larger for higher levels than for lower ones.

A counter-based flooding approach coupled with storage strategies is used to disseminate the data and to control the replication rate based on the importance level of the user information. Non-localized and localized storage algorithms are studied. For each storage mode, hop-counter and probabilistic replication strategies have been proposed. Previous dissemination algorithms (RaWMS, Supple, and DEEP), known to ensure uniform distribution, were used as a reference point for comparison. These algorithms were evaluated in terms of replication uniformity and communication overhead by using Castalia/OMNeT++ simulations. The simulation results showed that our probabilistic-based storage strategy, DEEP, and Supple provide better uniformity performance than our hop-counter-based algorithm and RaWMS. However, our probabilistic-based storage strategy, DEEP, and Supple incur the highest memory overhead.

Table 5
Performance synthesis of the evaluated algorithms.

	Storage strategy	Transmission strategy	Uniformity performance	Replication rate	Communication overhead
Proposed probabilistic algorithm	Probabilistic	Counter-based flooding	High	High	Low
Proposed hop-counter algorithm	Hop counter	Counter-based flooding	Low	Medium	Low
DEEP	Probabilistic	Probabilistic flooding	High	High	High
Supple	Probabilistic	Hierarchical tree	Medium	Medium	Low
RaWMS	Hop counter	Random walk	Low	Low	Medium

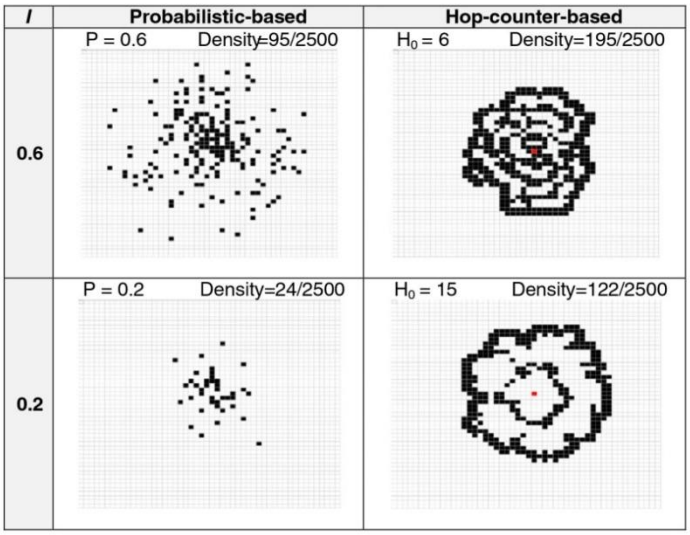


Fig. 22. Localized hop-counter storage of non-segmented data item.

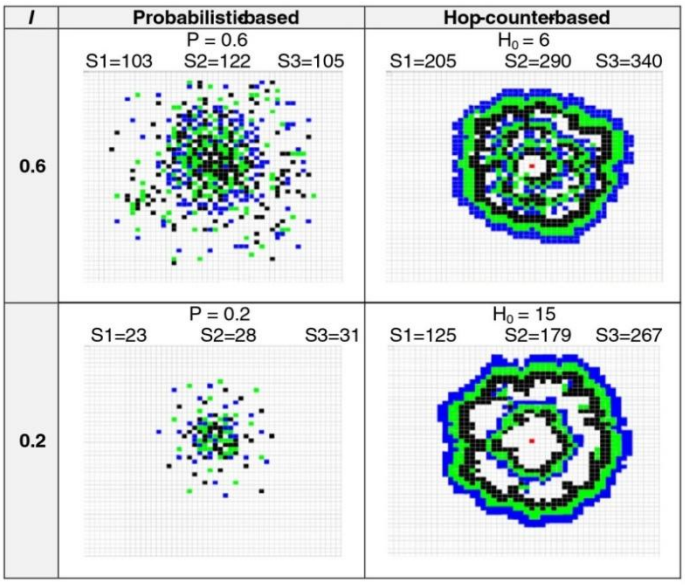


Fig. 23. Localized hop-counter storage of segmented data item.

Further, our algorithms and Supple have the lowest communication overhead. They result in low energy consumption of sensor nodes for all the data importance levels.

For localized dissemination, the simulation results showed an efficient localized storage for non-segmented data when the hop-counter and the probabilistic approaches were used. However, our probabilistic algorithm outperforms the other algorithms in the case of segmented items because all the segments are replicated in the target area with the same density.

Our future work will focus on testing the proposed solutions in actual concrete brick applications. The radio signal propagation inside such real material will be evaluated. Various performance metrics such as lost message ratio and average dissemination delay will be studied to evaluate their impact on replication uniformity and energy consumption.

References

[1] S. Kubler, W. Derigent, A. Thomas, E. Rondeau, Problem definition methodology for the communicating material paradigm, 10th IFAC Workshop on Intelligent Manufacturing Systems, Lisbon, Portugal2010.

[2] S. Kubler, W. Derigent, E. Rondeau, A. Thomas, Storage of information on manufactured products using communicating materials, 1st International Workshop on AI Problems and Approaches for Intelligent Environments, Montpellier, France2012 9–14.

[3] S. Kubler, W. Derigent, A. Thomas, E. Rondeau, Information dissemination process for context-aware products, 14th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'2012), Bucharest, Romania2012.

[4] S. Kubler, W. Derigent, A. Thomas, E. Rondeau, Embedding data on communicating materials from context-sensitive information analysis, J. Intell. Manuf. 25 (5) (2014) 1053–1064.

[5] S. Kubler, W. Derigent, A. Voisin, A. Thomas, E. Rondeau, Method for embedding context-sensitive information on communicating textiles via fuzzy AHP, J. Intell. Fuzzy Syst. 26 (2) (2014) 597–610.

[6] S. Kubler, A. Voisin, W. Derigent, A. Thomas, E. Rondeau, K. Främling, Group fuzzy AHP approach to embed relevant data on communicating material, J. Comput. Ind. 65 (4) (2014) 675–692.

[7] N. Krommenacker, P. Charpentier, R. Pena, Towards product health monitoring throughout its life-cycle, 11th IFAC Workshop on Intelligent Manufacturing System, Sao Paulo, Brazil2013 169–173.

[8] K. Bicakci, I.E. Bagci, B. Tavli, Z. Pala, Neighbor sensor networks: increasing lifetime and eliminating partitioning through cooperation, J. Comput. Stand. Interfaces 35 (4) (2013) 396–402.

[9] C. Ceken, An energy efficient and delay sensitive centralized MAC protocol for wireless sensor networks, J. Comput. Stand. Interfaces 30 (1-2) (2008) 20–31.

[10] A. Nechibvute, A. Chawanda, P. Luhanga, Piezoelectric energy harvesting devices: an alternative energy source for wireless sensors, J. Smart Mater. Res. (2012) 1–13.

[11] F. Hongping, F. Kangling, Overview of data dissemination strategy in wireless sensor networks, International Conference on E-Health Networking, Digital Ecosystems and Technologies, Shenzhen, China 2010, pp. 260–263.

[12] A. Cuevas, M. Uruena, G.D. Veciana, Dynamic random replication for data centric storage, 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, Bodrum, Turkey 2010, pp. 393–402.

[13] C.J.M. Liang, R. Musaloiu-E, A. Terzis, Typhoon: a reliable data dissemination protocol for wireless sensor networks, Comput. Sci. J. (2008) 268–285.

[14] M. Doddavenkatappa, M.C. Chan, B. Leong, Splash: fast data dissemination with constructive interference in wireless sensor networks, 10th USENIX Conference on Networked Systems Design and Implementation, Lombard, USA 2013, pp. 269–282.

[15] P. Kyasanur, R. Choudhury, I. Gupta, Smart gossip: an adaptive gossip-based broadcasting service for sensor networks, IEEE International Conference on Mobile Ad Hoc and Sensor Systems, Vancouver, British Columbia, Canada 2006, pp. 91–100.

[16] E.B. Hamida, G. Chelius, A line-based data dissemination protocol for wireless sensor networks with mobile sink, IEEE International Conference on Communications (ICC'08), Beijing, China, 2008.

[17] Y.J. Joung, S.H. Huang, Tug-of-war: an adaptive and cost-optimal data storage and query mechanism in wireless sensor networks, J. Distrib. Comput. Sensor Syst. 5067 (2008) 237–251.

[18] A.C. Rumín, M.U. Pascual, R.R. Ortega, D.L. López, Data centric storage technologies: analysis and enhancement, Sensors J. 10 (4) (2010) 3023–3056.

[19] S.S. Kulkarni, L. Wang, MNP: multihop network reprogramming service for sensor networks, 25th IEEE International Conference on Distributed Computing Systems, Columbus, Ohio, USA 2005, pp. 7–16.

[20] S. Saginbekov, A. Jhumka, Efficient code dissemination in wireless sensor networks, Futur. Gener. Comput. Syst. 39 (2014) 111–119.

[21] Y. Busnel, A. Ghodsi, K. Iwanicki, H. Miranda, H. Weatherspoon, Y. Busnel, A. Ghodsi, K. Iwanicki, H. Miranda, H. Weatherspoon, Gossiping over storage systems is practical, ACM SIGOPS Operating Systems Review41 (5)2007 75–81.

[22] O. Sekkas, D. Piguet, C. Anagnostopoulos, D. Kotsakos, G. Alyfantis, C. Kassapoglou-Faist, S. Hadjiethymiades, Probabilistic information dissemination for MANETs: the IPAC approach, 20th Tyrrhenian Workshop on Digital Communication, Pula, Italy2010 375–385.

[23] P. Kyasanur, R. Choudhury, I. Gupta, Smart gossip: an adaptive gossip-based broadcasting service for sensor networks, IEEE International Conference on Mobile Ad Hoc and Sensor Systems, Vancouver, British Columbia, Canada 2006, pp. 91–100.

[24] H. Miranda, S. Leggio, L. Rodrigues, K. Raatikainen, A power-aware broadcasting algorithm, 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Helsinki, Finland2006 1–5.

[25] C. Zhi-yan, J. Zhen-zhou, H. Ming-zeng, An energy-aware broadcast scheme for directed diffusion in wireless sensor network, J. Commun. Comput. 4 (5) (2007) 28–35.

[26] E.R. Sanchez, M. Rebaudengo, L. Zhang, Performance evaluation of reliable and unreliable opportunistic flooding in wireless sensor network, 17th IEEE International Conference on Networks, Singapore 2011, pp. 7–12.

[27] B. Garbinato, D. Rochat, M. Tomassini, F. Vessaz, Injecting power-awareness into epidemic information dissemination in sensor networks, J. Futur. Gener. Comput. Syst. 26 (2010) 868–876.

[28] S. Izumi, T. Matsuda, H. Kawaguchi, C. Ohta, M. Yoshimoto, Improvement of counter-based broadcasting by random assessment delay extension for wireless sensor networks, IEEE International Conference on Sensor Technologies and Applications, Valencia, Spain 2007, pp. 76–81.

[29] G. Maia, D.L. Guidoni, A.C. Viana, A.L. Aquino, R.A. Mini, A.A. Loureiro, A distributed data storage protocol for heterogeneous wireless sensor networks with mobile sinks, Ad Hoc Netw. 11 (5) (2013) 1588–1602.

[30] P. Gonizzi, G. Ferrari, V. Gay, J. Leguay, Data dissemination scheme for distributed storage for IoT observation systems at large scale, J. Inf. Fusion 22 (2015) 16–25.

[31] M. Vecchio, A.C. Viana, A. Ziviani, R. Friedman, DEEP: Density based proactive data dissemination protocol for wireless sensor networks with uncontrolled sink mobility, J. Comput. Commun. 33 (8) (2010) 929–939.

[32] A.C. Viana, T. Herault, T. Largillier, S. Peyronnet, F. Zaïdi, Supple: a flexible probabilistic data dissemination protocol for wireless sensor networks, 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, Bodrum, Turkey 2010, pp. 385–392.

[33] Z. Bar-Yossef, R. Friedman, G. Kliot, RaWMS: randomwalk based lightweight membership service for wireless ad hoc networks, ACM Trans. Comput. Syst. 26 (2) (2008) 1–66.

[34] Y. Tseng, S. Ni, E. Shih, Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network, IEEE Trans. Comput. 52 (5) (2003) 545–557.

[35] C. Chen, C. Hsu, H. Wang, DISCOUNT: a hybrid broadcast scheme for wireless ad hoc networks, 62nd IEEE Conference on Vehicular Technology 2005, pp. 2706–2710.

[36] Retransmission and back-off strategies for broadcasting in multi-hop wireless networks, in: J. Arango, A. Efrat, S. Ramasubramanian, M. Krunz, S. Pink (Eds.),3rd IEEE International Conference on Broadband Communication, Networks and Systems, San José, California, USA, 2006.

[37] S. Izumi, T. Takeuchi, T. Matsuda, H. Kawaguchi, C. Ohta, M. Yoshimoto, Counter-based broadcasting with hop count aware random assessment delay extension for wireless sensor networks, IEICE Trans. Commun. E91-B (11) (2008) 3489–3498.

[38] K. Mekki, W. Derigent, A. Zouinkhi, E. Rondeau, M.N. Abdelkrim, Improvement of counter-based broadcasting scheme for long communicating material lifetime, Proceedings Engineering & Technology (PET) of IPCO Publisher, International Conference on Automation, Control, Engineering and Computer Science (ACECS'14), Monastir, Tunisia 2014, pp. 53–59.

[39] H. Sundani, H. Li, V. Devabhaktuni, M. Alam, P. Bhattacharya, Wireless sensor network simulators, a survey and comparisons, J. Comput. Networks 2 (5) (2012) 249–265.

[40] B. Musznicki, P. Zwierzykowski, Survey of simulators for wireless sensor networks, Int. J. Grid Distrib. Comput. 5 (3) (2012) 23–50.

[41] S. Harte, B. O'Flynn, R.V. Martínez-Català, E.M. Popovici, Design and implementation of a miniaturised, low power wireless sensor node, 18th European Conference on Circuit Theory and Design, Seville, Spain, 2007.

[42] S. Harte, B. O'Flynn, R.V. Martínez-Català, E. Popovici, Wireless sensor node design for heterogeneous networks, 32nd International Conference on Microelectronics and Packaging, Warszawa–Pułtusk, Poland, 2008.

[43] T.V. Dam, K. Langendoen, An adaptive energy-efficient MAC protocol for wireless sensor networks, 1st International Conference on Embedded Networked Sensor Systems, Los Angeles, California, USA 2003, pp. 171–180.

[44] F. Marcelloni, M. Vecchio, A simple algorithm for data compression in wireless sensor networks, IEEE Commun. Lett. 12 (6) (2008) 411–413.

[45] C.M. Sadler, M. Martonosi, Data compression algorithms for energy-constrained devices in delay tolerant networks, 4th International Conference on Embedded Networked Sensor Systems, Boulder, Colorado, USA 2006, pp. 265–278.

Liste des publications

Revues internationales avec comité de lecture

[R1] K. Mekki, W. Derigent, A. Zouinkhi, E. Rondeau, A. Thomas, M.N. Abdelkrim, "Non-localized and localized data storage in large-scale communicating materials: Probabilistic and hop-counter approaches", Computer Standards & Interfaces, Vol. 44, February 2016, pp. 243-257.

[R2] K. Mekki, A. Zouinkhi, W. Derigent, E. Rondeau, A. Thomas, M.N. Abdelkrim, "USEE: A uniform data dissemination and energy efficient protocol for communicating materials", Future Generation Computer Systems, Vol. 56, March 2016, pp. 651-663.

[R3] K. Mekki, W. Derigent, A. Zouinkhi, E. Rondeau, A. Thomas, M.N. Abdelkrim, "RaWPG: A data retrieval protocol in micro-sensor networks based on random walk and pull gossip for communicating materials", IEEE Internet of Things Journal. (Accepted)

Conférences internationales avec comité de lecture

[C1] K. Mekki, W. Derigent, A. Zouinkhi, E. Rondeau, M.N. Abdelkrim, "Data Dissemination Algorithms for Communicating Materials using Wireless Sensor Networks", The 2nd IEEE International Conference on Future Internet of Things and Cloud (FiCloud'2014), 27-29 August 2014, Barcelona, Spain. (Best Paper Award)

[C2] K. Mekki, W. Derigent, A. Zouinkhi, E. Rondeau, M.N. Abdelkrim, "An Uniform Data Replication Algorithm in Wireless Micro-Sensors Network for Communicating Materials Application", 10th International Conference on Modeling, Optimization & Simulation (MOSIM'2014), 5-7 November 2014, Nancy, France.

[C3] K. Mekki, W. Derigent, A. Zouinkhi, E. Rondeau, M.N. Abdelkrim, "Multi-hop Counter Based Data Dissemination Protocol for Communicating Materials", 9th IEEE International Conference on Wireless and Mobile Computing, Networking, and Communications (WiMob'2013), 7-9 October 2013, Lyon, France.

[C4] K. Mekki, W. Derigent, A. Zouinkhi, E. Rondeau, M.N. Abdelkrim, "Improvement of Counter-Based Broadcasting Scheme for Long Communicating Material Lifetime", International Conference on Automation, Control, Engineering & Computer Science (ACECS'2014), 22-25 March 2014, Monastir, Tunisia.

Bibliographie

(Ahvar and Mahdavi, 2011) S. Ahvar, M. Mahdavi, "EEQR: An Energy Efficient Query-Based Routing Protocol for Wireless Sensor Networks", Journal of Advances in Computer research, Vol. 2, No. 3, 2011, pp 25-38.

(Ahvar et al., 2012) S. Ahvar, M. Mahdavi, E. Ahvar, "Improving EEQR Algorithm in Wireless Sensor Networks", 6th International Symposium on Telecommunications, November 2012, Tehran.

(Albin, 2014) J. Albin, "Introduction du concept de matière communicante dans la construction béton", Master Thesis ISC, Université de Lorraine, 2014, Nancy, France.

(Aminu et al., 2007) M. Aminu, M. Ould-Khaoua, L. Mackenzie, "An Efficient Counter-Based Broadcast Scheme for Mobile Ad Hoc Networks", 4th European Performance Engineering Workshop, September 2007, Berlin, Germany.

(Anastasi et al., 2009) G. Anastasi, M. Conti, M.D. Francesco, A. Passarella, "Energy conservation in wireless sensor networks: A survey", Ad Hoc Networks, Vol. 7, 2009, pp. 537-568.

(Arango et al., 2006) J. Arango, A. Efrat, S. Ramasubramanian, M. Krunz, S. Pink, "Retransmission and Back-off Strategies for Broadcasting in Multi-hop Wireless Networks", 3rd IEEE International Conference on Broadband Communication, Networks and Systems, October 2006, San José, California, USA.

(Ash and Potter, 2004) J.N. Ash, L.C. Potter, "Sensor network localization via received signal strength measurements with directional antennas", 42nd Allerton Conference on Communication, Control, and Computing, September 2004, Monticello, USA.

(Bar-Yossef et al., 2008) Z. Bar-Yossef, R. Friedman, G. Kliot, "RaWMS: RandomWalk based Lightweight Membership Service for Wireless Ad Hoc Networks", ACM Transactions on Computer Systems, Vol. 26, No. 2, 2008, pp. 1-66.

(Beigl et al., 2005) M. Beigl, C. Decker, A. Krohn, T. Riedel, T. Zimmer, "µParts: Low Cost Sensor Networks at Scale", 7th International Conference on Ubiquitous Computing, September 2005, Tokyo, Japan.

(Bohn et al., 2005) J. Bohn, V. Coroama, M. Langheinrich, F. Mattern, M. Rohs, "Social, economic, and ethical implications of ambient intelligence and ubiquitous computing", Ambient Intelligence, 2005, pp. 5-29.

(Boulis, 2011) A. Boulis, "Castalia, a simulator for wireless sensor networks and body area networks, version 3.2", User's manual, NICTA, March 2011.

(Braginsky and Estrin, 2002) D. Braginsky, D. Estrin, "Rumor Routing Algorithm for Sensor Networks", 1st Workshop on Sensor Networks and Applications, October 2002, Atlanta, USA.

(Buschmann et al., 2006) C. Buschmann, D. Pfisterer, S. Fischer, "Estimating distances using neighborhood intersection", IEEE Conference on Emerging Technologies and Factory Automation, September 2006, Prague, Czech.

(Buschmann et al., 2007) C. Buschmann, H. Hellbruck, S. Fischer, A. Kroller, S. Fekete, "Radio propagation-aware distance estimation based on neighborhood comparison", 4th European conference on Wireless sensor networks, January 2007, Delft, Netherlands.

(Busnel et al., 2007) Y. Busnel, A. Ghodsi, K. Iwanicki, H. Miranda, H. Weatherspoon, "Gossiping over storage systems is practical", ACM SIGOPS Operating Systems Review, Vol. 41, No. 5, 2007, pp. 75-81.

(Chang et al., 2005) C. Chang, K. Shih, S. Lee, "ZBP: A Zone-Based Broadcasting Protocol for Wireless Sensor Networks", Journal of Wireless Personal Communications, Vol. 33, 2005, pp. 53-68.

(Chao et al., 2007) C. Chao, J. Yang, W. Jen, "Determining technology trends and forecasts of RFID by a historical review and bibliometric analysis from 1991 to 2005", Technovation, Vol. 27, No. 5, 2007, pp. 268-279.

(Cheng et al., 2009) L. Cheng, Y. Chen, C. Chen, J. Ma, "Query-based data collection in wireless sensor networks with mobile sinks", International Conference on Wireless Communications and Mobile Computing, June 2009, New York, USA.

(Cimen et al., 2003) C. Cimen, E. Cayirci, V. Coskun, "Querying Sensor Fields by using Quadtree based Dynamic Cluster and Task Sets", IEEE Conference on Military Communications, October 2003, Boston, Massachusetts, USA.

(Couto et al., 2005) D.S.J.D. Couto, D. Aguayo, J. Bicket, R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing", Wireless Network Journal, Vol. 11, 2005, pp. 419-434.

(Das et al., 2005) S. Das, H. Pucha, Y. Hu, "Performance comparison of scalable location services for geographic ad hoc routing", 24th Annual IEEE Conference of the IEEE Computer and Communications Societies, March 2005, Miami, USA.

(Doss et al., 2010) R. Doss, G. Li, V. Mark, M. Tissera, "Information discovery in mission-critical wireless sensor networks", Computer Networks Journal, Vol. 54, 2010, pp. 2383-2399.

(Drabkin et al., 2007) V. Drabkin, R. Friedman, G. Kliot, M. Segal, "RAPID: reliable probabilistic dissemination in wireless ad hoc networks", 26th IEEE International Symposium on Reliable Distributed Systems, October 2007, Beijing, China.

(ElFatni et al., 2011) A. ElFatni, G. Juanole, T. Val, "Modélisation et analyse des protocoles MAC multi-canaux basés sur l'approche du canal dédié", CFIP - Colloque Francophone sur l'Ingénierie des Protocoles, May 2011, Sainte Maxime, France.

(Ellen MacArthur, 2016) Ellen MacArthur, "Intelligent Assets: Unlocking the circular economy potential", Rapport de la fondation Ellen MacArthur, Février 2016.

(Erman et al., 2012) A. Erman, A. Dilo, P. Havinga, "A virtual infrastructure based on honeycomb tessellation for data dissemination in multi-sink mobile wireless sensor networks", EURASIP Journal on Wireless Communications and Networking, Vol. 2012, No. 1, December 2012, pp. 1-27.

(Finkenzeller, 2003) K. Finkenzeller, "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification", Second Edition, 2003, Wiley, ISBN: 0-470-84402-7.

(Ganesan et al., 2001) D. Ganesan, R. Govindan, S. Shenker, D. Estrin, "Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks", Mobile Computer Communication Review, Vol. 5, 2001, pp. 11-25.

(Garbinato et al., 2010) B. Garbinato, D. Rochat, M. Tomassini, F. Vessaz, "Injecting power-awareness into epidemic information dissemination in sensor networks", Journal of Future Generation Computer Systems, Vol. 26, 2010, pp. 868-876.

(Gauthier and Regourd, 1982) E. Gauthier, M. Regourd, "Le durcissement du béton en fonction de la température", Colloque international sur le béton jeune, 6-7 Avril 1982, Paris, France.

(Gershenfeld et al., 2004) N. Gershenfeld, R. Krikorian, D. Cohen, "The internet of things", Scientific American, Vol. 291, No. 4, 2004, pp.76-81.

(Grosse et al., 2006) C.U. Grosse, S.D. Glaser, M. Krüger, "Condition monitoring of concrete structures using wireless sensor networks and MEMS", Smart Structures and Materials 2006: Smart Structures and Integrated Systems, Vol. 617302, April 2006.

(Gu and Stankovic, 2005) L. Gu, J.A. Stankovic, "Radio-Triggered Wake-up for Wireless Sensor Networks", Real-Time Systems, Vol. 29, No. 2, March 2005, pp. 157-182.

(Guo and Zhang, 2014) W. Guo, W. Zhang, "A survey on intelligent routing protocols in wireless sensor networks", Journal of Network and Computer Applications, Vol. 38, 2014, pp. 185-201.

(Hakoura, 2011) B.Z. Hakoura, "Comparison of Collection Tree Protocols with Gossip Algorithms for Data Aggregation in Wireless Sensor Networks", Master thesis of Engineering, September 2011, McGill University, Montreal, Canada.

(Hamida and Chelius, 2008) E.B. Hamida, G. Chelius, "A line-based data dissemination protocol for wireless sensor networks with mobile sink", IEEE International Conference on Communications, May 2008, Beijing, China.

(Harper, 2003) R. Harper, "Inside the Smart Home", First Edition, Springer-Verlag London, 2003, ISBN: 978-1-85233-854-1.

(Harte et al., 2008) S. Harte, B. O'Flynn, R.V. Martínez-Català, E. Popovici, "Wireless sensor node design for heterogeneous networks", 32nd International Conference on Microelectronics and Packaging, September 2008, Pułtusk, Poland.

(Hassanein and Luo, 2006) H. Hassanein, J. Luo, "Reliable Energy Aware Routing in Wireless Sensor Networks", 2nd IEEE Workshop on Dependability and Security in Sensor Networks and Systems, April 2006, Los Alamitos, CA, USA.

(Huang et al., 2014) B. Huang, C. Yu, B.D.O. Anderson, G. Mao, "Estimating Distances via Connectivity in Wireless Sensor Networks", Journal of Wireless Communications and Mobile Computing, Vol. 14, No. 5, April 2014, pp. 541-556.

(Intanagonwiwat et al., 2000) C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", 6th Annual International Conference on Mobile Computing and Networking, August 2000, Boston, MA, USA.

(Ives and Vitale, 1988) B. Ives, M. Vitale, "After the sale: leveraging maintenance with information technology", MIS Quarterly, Vol. 12, No. 1, 1988, pp. 7-21.

(Izumi et al., 2007) S. Izumi, T. Matsuda, H. Kawaguchi, C. Ohta, M. Yoshimoto, "Improvement of Counter-based Broadcasting by Random Assessment Delay Extension for Wireless Sensor Networks", IEEE International Conference on Sensor Technologies and Applications, October 2007, Valencia, Spain.

(Izumi et al., 2008) S. Izumi, T. Takeuchi, T. Matsuda, H. Kawaguchi, C. Ohta, M. Yoshimoto, "Counter-Based Broadcasting with Hop Count Aware Random Assessment Delay Extension for Wireless Sensor Networks", IEICE Transactions on Communications, Vol. E91-B, No. 11, November 2008, pp. 3489-3498.

(Jaber, 2009) M. Jaber, "Architecture de Système d'Information Distribué pour la Gestion de la Chaîne Logis- tique : Une approche Orientée Services", PhD thesis, Institut National des Sciences Appliquées de Lyon, 2009, Lyon, France.

(Jacobsson et al., 2011) M. Jacobsson, C. Guo, I. Niemegeers, "An experimental investigation of optimized flooding protocols using a wireless sensor network testbed", Journal of Computer Networks, Vol. 55, 2011, pp. 2899-2913.

(Jansen-Vullers et al., 2003) M. H. Jansen-Vullers, C. A. Van Dorp, A. J. M Beulensb, "Managing traceability information in manufacture", International Journal of Information Management, Vol. 23, 2003, pp. 395-413.

(Jelicic et al., 2012) V. Jelicic, M. Magno, D. Brunelli, V. Bilas, L. Benini, "Analytic comparison of wake-up receivers for WSNs and benefits over the wake-on radio scheme", 7th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks, October 2012, Paphos, Cyprus.

(Jover et al., 2011) J. Jover, A. Thomas, V. Bombardier, "Marquage du bois dans la masse : intérêts et perspectives", 9ème Congrès International de Génie Industriel, October 2011, Saint-Sauveur, Québec, Canada.

(Jover, 2013) Jeremy Jover, "Contribution à la réduction des pertes d'informations dans l'industrie du bois", PhD Thesis, école doctorale IAEM, université de Lorraine, 2013, Nancy, France.

(Jover et al., 2013) J. Jover, A. Thomas, J. M. Leban, D. Canet, "Interest of new communicating material paradigm: An attempt in wood industry", Journal of Physics: Conference Series, Vol. 416, No. 1, 2013, pp. 12-31.

(Kalansuriya et al., 2013) P. Kalansuriya, B. Rahul, S. Sanjay, "RFID Tag Antenna-Based Sensing for Pervasive Surface Crack Detection", IEEE Sensors Journal, Vol. 13, No. 5, 2013, pp. 1564-1570.

(Khiati and Djenouri, 2012) M. Khiati, D. Djenouri, "Cluster-Based Fast Broadcast in Duty-Cycled Wireless Sensor Networks", 11th IEEE International Symposium on Network Computing and Applications, August 2012, Cambridge, Massachusetts, USA.

(Krommenacker et al., 2013) N. Krommenacker, P. Charpentier, R. Peña, "Towards Product Health Monitoring throughout its life-cycle", 11th IFAC Workshop on Intelligent Manufacturing System, May 2013, Sao Paulo, Brazil.

(Kubler et al., 2011) S. Kubler, W. Derigent, A. Thomas, E. Rondeau, "Prototyping of a communicating textile", International Conference on Industrial Engineering and Systems Management, May 2011, Metz, France

(Kubler, 2012) S. Kubler, "Premiers travaux relatifs au concept de matière communicante", PhD Thesis, école doctorale IAEM, université de Lorraine, 2012, Nancy, France.

(Kubler et al., 2012a) S. Kubler, W. Derigent, E. Rondeau, A. Thomas, "Storage of information on manufactured products using communicating materials", 1st International Workshop on AI Problems and Approaches for Intelligent Environments, August 2012, Montpellier, France.

(Kubler et al., 2012b) S. Kubler, W. Derigent, A. Thomas, E. Rondeau, "Information dissemination process for context-aware products", 14th IFAC Symposium on Information Control Problems in Manufacturing, May 2012, Bucharest, Romania.

(Kubler et al., 2014a) S. Kubler, W. Derigent, A. Thomas, E. Rondeau, "Embedding data on communicating materials from context-sensitive information analysis", Journal of Intelligent Manufacturing, Vol. 25, No. 5, October 2014, pp. 1053-1064.

(Kubler et al., 2014b) S. Kubler, A. Voisin, W. Derigent, A. Thomas, E. Rondeau, K. Främling, "Group fuzzy AHP approach to embed relevant data on communicating material", Journal of Computers in Industry, Vol. 65, No. 4, 2014, pp. 675-692.

(LAFARGE, 2013) LAFARGE, "Quand les ntic s'invitent sur le chantier de la tour d2(92): Une collaboration réussie entre Lafarge et Vinci Construction France", Communiqué de presse, Paris, 6 septembre 2013.

(Lee et al., 2008) L. Lee, K. Fiedler, J. Smith, "Radio frequency identification (RFID) implementation in the service sector: A customer-facing diffusion model", International Journal of Production Economics, Vol. 112, No. 2, 2008, pp. 587-600.

(Li et al., 2010) S. Li, R.K. Neelisetti, C. Liu, A. Lim, "Efficient Multi-Path protocol for Wireless Sensor Networks", International journal of Wireless Mobile Network, Vol. 2, 2010, pp. 110-130.

(Lima and Barros, 2007) L. Lima, L. Barros, "Random Walks on Sensor Networks", 5th IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, April 2007, Limassol, Chypre.

(Liu et al., 2007) X. Liu, Q. Huang, Y. Zhang, "Balancing push and pull for efficient information discovery in large-scale sensor networks", IEEE Transactions on Mobile Computing, Vol. 6, No. 3, 2007, pp. 241-257.

(Lou, 2005) W. Lou, "An Efficient N-to-1 Multipath Routing Protocol in Wireless Sensor Networks", 2nd IEEE International Conference on Mobile Ad-hoc and Sensor System, November 2005, Washington, DC, USA.

(Lu et al., 2005) C. Lu, G. Xing, O. Chipara, C.-L. Fok, S. Bhattacharya, "A spatiotemporal query service for mobile users in sensor networks", 25th IEEE International Conference on Distributed Computing Systems, June 2005, Columbus, Ohio, USA.

(Luo et al., 2005) H. Luo, F. Ye, J. Cheng, S. Lu, L. Zhang, "TTDD: Two-Tier Data Dissemination in Large-Scale Wireless Sensor Networks", Wireless Networks Journal, Vol. 11, 2005, pp. 161-175.

(Machado, 2005) M. Machado, "Data dissemination in autonomic wireless sensor networks", IEEE Journal on Selected Areas in Communications, Vol. 23, No. 12, 2005, pp. 2305-2319.

(Magno et al., 2014) M. Magno, S. Marinkovic, B. Srbinovski, E.M. Popovici, "Wake-up radio receiver based power minimization techniques for wireless sensor networks: A review", Microelectronics Journal, Vol. 45, No. 12, December 2014, pp. 1627-1633.

(Maimour, 2008) M. Maimour, "Maximally Radio-Disjoint Multipath Routing for Wireless Multimedia Sensor Networks", 4th ACM Workshop on Wireless Multimedia Networking and Performance Modeling, October 2008, Vancouver, BC, Canada.

(Mao et al., 2007) G. Mao, B. Fidan, B.D.O. Anderson, "Wireless sensor network localization techniques", Computer Networks, Vol. 51, No. 10, July 2007, pp. 2529-2553.

(Marcelloni and Vecchio, 2008) F. Marcelloni, M. Vecchio, "A simple algorithm for data compression in wireless sensor networks", IEEE Communications Letters, Vol. 12, No. 6, 2008, pp. 411-413.

(Maria et al., 2013) G. Maria, D.L. Guidoni, A.C. Viana, A.L.L. Aquino, R.A.F. Mini, A.A.F. Loureiro, "A distributed data storage protocol for heterogeneous wireless sensor networks with mobile sinks", Ad Hoc Networks Journal, Vol. 11, 2013, pp. 1588-1602.

(Marinkovic and Popovici, 2011) S. Marinkovic, E. Popovici, "Nano-Power Wireless Wake-Up Receiver With Serial Peripheral Interface", IEEE Journal on Selected Areas in Communications (Vol. 29, No. 8, September 2011, pp. 1641-1647.

(McFarlane et al., 2003) D. McFarlane, S. Sarma, J. L. Chirn, C.Y. Wong, K. Ashton, "Auto id systems and intelligent manufacturing control", Engineering Applications of Artificial Intelligence, Vol. 16, No. 4, 2003, pp. 365-376.

(McFarlane et al., 2012) D. McFarlane, V. Giannikas, A.C. Wong, M. Harrison, "Intelligent products in the supply chain - 10 years on", 14th Symposium of Information Control Problems in Manufacturing, 2012, Bucharest, Romania.

(Mekki et al., 2013) K. Mekki, W. Derigent, A. Zouinkhi, E. Rondeau, M.N. Abdelkrim, "Multi-hop Counter Based Data Dissemination Protocol for Communicating Materials", 9th IEEE International Conference on Wireless and Mobile Computing, Networking, and Communications, October 2013, Lyon, France.

(Mekki et al., 2014a) K. Mekki, W. Derigent, A. Zouinkhi, E. Rondeau, M.N. Abdelkrim, "Data Dissemination Algorithms for Communicating Materials using Wireless Sensor Networks", 2nd IEEE International Conference on Future Internet of Things and Cloud, August 2014, Barcelona, Spain.

(Mekki et al., 2014b) K. Mekki, W. Derigent, A. Zouinkhi, E. Rondeau, M.N. Abdelkrim, "Improvement of Counter-Based Broadcasting Scheme for Long Communicating Material Lifetime", International Conference on Automation, Control, Engineering & Computer Science, March 2014, Monastir, Tunisia.

(Mekki et al., 2016a) K. Mekki, W. Derigent, A. Zouinkhi, E. Rondeau, A. Thomas, M.N. Abdelkrim, "Non-Localized and Localized Data Storage in Large-Scale Communicating Materials: Probabilistic

and Hop-Counter Approaches", Journal of Computer Standards & Interfaces, Vol. 44, February 2016, pp. 243-257.

(Mekki et al., 2016b) K. Mekki, W. Derigent, A. Zouinhki, E. Rondeau, A. Thomas, M.N. Abdelkrim, "USEE: A Uniform Data Dissemination and Energy Efficient Protocol for Communicating Materials", Journal of Future Generation Computer Systems, Vol. 56, March 2016, pp. 651-663.

(Mir and Ko, 2006) Z.H. Mir, Y.-B. Ko, "A Quadtree-Based Data Dissemination Protocol for Wireless Sensor Networks with Mobile Sinks", 11th International Conference on Personal Wireless Communications, September 2006, Albacete, Spain.

(Morvaj et al., 2011) B. Morvaj, L. Lugaric, S. Krajcar, "Demonstrating smart buildings and smart grid features in a smart energy city", 3rd International Youth Conference on Energetics, July 2011, Leiria, Portugal.

(Musilek et al., 2015) P. Musilek, P. Kromer, T. Barton, "Review of nature-inspired methods for wake-up scheduling in wireless sensor networks", Swarm and Evolutionary Computation, Vol. 25, 2015, pp. 100-118.

(Musznicki and Zwierzykowski, 2012) B. Musznicki, P. Zwierzykowski, "Survey of Simulators for Wireless Sensor Networks", International Journal of Grid and Distributed Computing, Vol. 5, No. 3, 2012, pp. 23-49.

(Nechibvute et al., 2012) A. Nechibvute, A. Chawanda, P. Luhanga, "Piezoelectric Energy Harvesting Devices: An Alternative Energy Source for Wireless Sensors", Journal of Smart Materials Research, Vol. 2012, 2012, pp. 1-13.

(Norman, 1998) D.A. Norman, "The Invisible Computer: Why Good Products can Fail, the Personal Computer is so Complex, and Information Appliances are the Solution", MIT Press, 1998.

(Othman and Yahya, 2010) J.B. Othman, B. Yahya, "Energy efficient and QoS based routing protocol for wireless sensor networks", Journal of Parallel and Distributed Computing, Vol. 70, No. 8, August 2010, pp. 849-857.

(Ouwerkerk et al., 2006) M. Ouwerkerk, F. Pasveer, N. Engin, "SAND: a modular application development platform for miniature wireless sensors", International Workshop on Wearable and Implantable Body Sensor Networks, April 2006, Cambridge, MA, USA.

(Park et al., 2005) C. Park, J. Liu, P.H. Chou, "Eco: an Ultra-compact Low-Power Wireless Sensor Node for Real-Time Motion Monitoring", 4th International Conference on Information Processing in Sensor Networks, April 2005, Los Angeles, CA, USA.

(Pediaditakis et al., 2010) D. Pediaditakis, Y. Tselishchev, A. Boulis, "Performance and Scalability Evaluation of the Castalia Wireless Sensor Network Simulator", 3rd International ICST Conference on Simulation Tools and Techniques, March 2010, Malaga, Spain.

(Peña et al., 2011) R. Peña, N. Krommenacker, P. Charpentier, "A New Strategy for Dimensional Metrology using Smart Dust Networks", International Conference on Indoor Positioning and Indoor Navigation, September 2011, Guimarães, Portugal.

(Peng and Sichitiu, 2006) R. Peng, M.L. Sichitiu, "Angle of arrival localization for wireless sensor networks", 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, September 2006, Reston, Virginia, USA.

(Rachuri et al., 2008) S. Rachuri, E. Subrahmanian, A. Bouras, S.J. Fenves, S. Foufou, R.D. Sriram, "Information sharing and exchange in the context of product lifecycle management: Role of standards", Computer-Aided Design, Vol. 40, 2008, pp. 789-800.

(Rashid and Rehmani, 2016) B. Rashid, M.H. Rehmani, "Applications of wireless sensor networks for urban areas: A survey", Journal of Network and Computer Applications, Vol. 60, January 2016, pp. 192-219.

(Rastegarnia et al., 2011) A. Rastegarnia, V. Solouk, "Performance evaluation of Castalia Wireless Sensor Network simulator", 34th IEEE International Conference on Telecommunications and Signal Processing, August 2011, Budapest, Hongrie.

(Sadler and Martonosi, 2006) C.M. Sadler, M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks", 4th International Conference on Embedded Networked Sensor Systems, 2006, Boulder, Colorado, USA.

(Sahin, 2004) Evren Sahin, "A qualitative and quantitative analysis of the impact of the Auto ID technology on the performance of supply chains", PhD Thesis, Ecole Centrale Paris, 2004, Paris, France.

(Sanchez et al., 2011) E.R. Sanchez, M. Rebaudengo, L. Zhang, "Performance evaluation of reliable and unreliable opportunistic flooding in wireless sensor network", 17th IEEE International Conference on Networks, December 2011, Singapore.

(Schurgers and Srivastava, 2001) C. Schurgers, M.B. Srivastava, "Energy efficient routing in wireless sensor networks", IEEE Military Communications Conference, October 2001, McLean, VA, USA.

(Sen and Ukil, 2009) J. Sen, A. Ukil, "An Adaptable and QoS-Aware Routing Protocol for Wireless Sensor Networks", 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, May 2009, Aalborg, Denmark.

(Shah and Rabaey, 2002) R.C. Shah, J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks", IEEE Conference on Wireless Communications and Networking, March 2002, Orlando, USA.

(Shi et al., 2012) G. Shi, J. Zheng, J. Yang, Z. Zhao, "Double-blind data discovery using double cross for large-scale wireless sensor networks with mobile sinks", IEEE Transactions on Vehicular Technology, Vol. 61, No. 5, 2012, pp. 2294-2304.

(Shih et al., 2011) W. Shih, R. Jurdak, B. Lee, D. Abbott, "High sensitivity wake-up radio using spreading codes: design, evaluation, and applications", EURASIP Journal on Wireless Communications and Networking, Vol. 26, December 2011.

(Shin et al., 2005) J.-H. Shin, J. Kim, K. Park, D. Park, "RailRoad: Virtual Infrastructure for Data Dissemination in Wireless Sensor Networks", 2nd ACM International Workshop on Performance Evaluation of Wireless Ad hoc, Sensor, and Ubiquitous Networks, October 2005, Montreal, Quebec, Canada.

(Shokrzadeh et al., 2007) H. Shokrzadeh, A. T. Haghighat, A. Nayebi, "Directional Rumor Routing in Wireless Sensor Networks", Third IEEE/IFIP International Conference in Central Asia on Internet, September 2007, Tachkent, Ouzbékistan.

(Singh et al., 2008) C.P. Singh, O.P. Vyas, M.K. Tiwari, "An Overview of Routing Protocols of Sensor Networks", IEEE International Conference on Computational Intelligence for Modelling Control & Automation, December 2008, Vienna, Austria.

(Song, 2013) Y. Song, "Réseaux de Capteurs Sans Fil : Comment Fournir La Qualité de Service Tout En Economisant l'Energie ?", Ecole d'été temps réel, August 2013, Toulouse, France.

(Tajima, 2007) M. Tajima, "Strategic value of RFID in supply chain management", Journal of purchasing and supply management, Vol. 13, No. 4, 2007, pp. 261-273.

(Thérèse, 2016) A.A. Thérèse, "Réseaux de capteurs sans fil étendus dédiés aux collectes de données environnementales", PhD Thesis, école doctorale sciences pour l'ingénieur de Clermont-Ferrand, Université Blaise Pascal - Clermont II, 2016, Aubière, France.

(Thomas, 2009) A. Thomas, "RFID et nouvelles technologies de communication : enjeux économiques incontournables et problèmes d'éthique", 6ème Conférence Internationale Conception et Production Integrées, Octobre 2009, Fès, Maroc.

(Tian et al., 2006) H. Tian, H. Shen, T. Matsuzawa, "Random Walk Routing in WSNs with Regular Topologies", Journal of Computer Science and Technology, Vol. 21, No. 4, July 2006, pp. 496-502.

(Torfs et al., 2004) T. Torfs, S. Sanders, C. Winters, S. Brebels, C.V. Hoof, "Wireless network of autonomous environmental sensors", 3rd IEEE Conference on Sensors, October 2004, Vienna, Austria.

(Tran-Dang et al., 2014) H. Tran-Dang, N. Krommenacker, P. Charpentier, "Localization algorithms based on Hop counting for wireless nano-sensor networks", 5th International Conference on Indoor Positioning and Indoor Navigation, October 2014, Busan, South Korea.

(Tunca et al., 2014) C. Tunca, S. Isik, M.Y. Donmez, C. Ersoy, "Distributed Mobile Sink Routing for Wireless Sensor Networks: A Survey", IEEE Communications Surveys & Tutorials, Vol. 16, No. 2, 2014, pp. 887-897.

(Vecchio et al., 2010) M. Vecchio, A.C. Viana, A. Ziviani, R. Friedman, "DEEP: Density based proactive data dissemination protocol for wireless sensor networks with uncontrolled sink mobility", Journal of Computer Communications, Vol. 33, No. 8, 2010, pp. 929-939.

(Viana et al., 2010) A.C. Viana, T. Herault, T. Largillier, S. Peyronnet, F. Zaïdi, "Supple: A Flexible Probabilistic Data Dissemination Protocol for Wireless Sensor Networks", 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, October 2010, Bodrum, Turkey.

(Vrijhoef, 2000) R. Vrijhoef, "The four roles of supply chain management in Construction", European Journal of Purchassing & Supply Management, Vol. 6, No. 3, 2000, pp. 169-178.

(Wan et al., 2011) Z.G. Wan, Y.K. Tan, C. Yuen, "Review on energy harvesting and energy management for sustainable wireless sensor networks", 13th IEEE International Conference on Communication Technology, September 2011, Jinan, China.

(Wang et al., 2009) G. Wang, T. Wang, W. Jia, M. Guo, J. Li, "Adaptive location updates for mobile sinks in wireless sensor networks", Journal of Supercomputing, Vol. 47, No. 2, 2009, pp. 127-145.

(Wohlers et al., 2009) R. Wohlers, N. Trigoni, R. Zhang, S. Ellwood, "TwinRoute: Energyefficient data collection in fixed sensor networks with mobile sinks", 10th IEEE International Conference on Mobile Data Management: Systems, Services and Middleware, May 2009, Taipei, Taiwan.

(Wong et al., 2002) C.Y. Wong, D. McFarlane, A. Zaharudin, V. Agarwal, "The Intelligent Product Driven Supply Chain", IEEE International Conference on Systems, Man and Cybernetics, October 2002, Hammamet, Tunisia.

(Yamashita et al., 2006) S. Yamashita, T. Shimura, K. Aiki, K. Ara, Y. Ogata, I. Shimokawa, T. Tanaka, H. Kuriyama, K. Shimada, K. Yano, "A 15 × 15 mm, 1 μA, Reliable Sensor-Net Module: Enabling Application-Specific Nodes", 5th International Conference on Information Processing in Sensor Networks, April 2006, Nashville, TN, USA.

(Ye et al., 2001) F. Ye, A. Chen, S. Lu, L. Zhang, "A scalable solution to minimum cost forwarding in large sensor networks", 10th International Conference on Computer Communications and Networks, October 2001, Scottsdale, AZ, USA.

(Ye et al., 2005) F. Ye, G. Zhong, S. Lu, L. Zhang, "Gradient broadcast: A robust data delivery protocol for large scale sensor networks", Wireless Networks, Vol. 11, 2005, pp. 285-298.

(Yubo et al., 2009) D. Yubo, N. Jianwei, L. Lian, "Modeling of Broadcasting Based on Distance Scheme for WSN", 5th International Joint Conference on INC, IMS and IDC, August 2009, Seoul, South Korea.

(Zaetang et al., 2016) Y. Zaetang, V. Sata, A. Wongsa, P. Chindaprasirt, "Properties of pervious concrete containing recycled concrete block aggregate and recycled concrete aggregate", Journal of Construction and Building Materials, Vol. 111, May 2016, pp. 15-21.

(Zheng, 2014) H. Zheng, "Auto-Configuration, Supervision et Contrôle d'Entités Physiques par l'intermédiaire de Réseaux de Capteurs et Actionneurs", PhD thesis, école doctorale : Informatique et Mathématiques de Lyon, Université de Lyon, 2014, France.

(Zouinkhi et al., 2011) A. Zouinkhi, E. Bajic, E. Rondeau, MN. Abdelkrim, "Simulation and modeling of active products cooperation for active security system management", International Journal of Transactions on Systems, Signals and Devices, Vol. 5, No. 3, 2011, pp. 1-23.

Résumé

La thèse aborde le problème de la dissémination des informations liées au produit tout au long de son cycle de vie, par l'exploitation du concept de matière communicante. L'objectif général est de stocker dans la matière communicante ses caractéristiques initiales mais aussi l'évolution de ses propriétés durant son usage, en évitant de les perdre lors par exemple d'une transformation ou d'une destruction d'une partie du matériau. Dans le cadre de cette thèse, ce principe est appliqué dans le domaine des préfabriqués en béton, où des nœuds de réseaux de capteurs sans fil sont intégrés dans le béton. Ces nœuds sont alors utilisés pour stocker des informations relatives au cycle de vie du préfabriqué et à sa surveillance sur la phase d'usage. Un nouveau protocole de communication, nommé USEE, a été proposé et permet de diffuser uniformément les informations dans la matière en considérant qu'elles n'ont pas toutes la même importance. Le protocole USEE évite notamment la saturation rapide des mémoires des nœuds de façon à pouvoir stocker un maximum d'informations différentes. Ensuite, un protocole de lecture, intitulé RaWPG, a été développé. Il est adapté à la récupération d'informations uniformément réparties et consomme peu d'énergie. Ces deux protocoles ont été implémentés dans le simulateur réseau Castalia/OMNeT++ et ont permis de montrer leur intérêt par rapport au contexte applicatif mais aussi par rapport à d'autres protocoles similaires de la littérature.

Mots clés

Internet des objets, gestion de cycle de vie des produits, matière communicante, réseaux de capteurs sans fil.

Abstract

A new Internet of Things area is coming with communicating materials, which are able to provide diverse functionalities to users all along the product lifecycle. As example, it can track its own evolution which leads to gather helpful information. This new paradigm is fulfilled via the integration of specific electronic components into the product material. In this thesis, ultra-small wireless sensor nodes are used for concrete precast field. Indeed, storage of lifecycle information and data dissemination in communicating materials are very important issues. Therefore, this thesis provides a new protocol (USEE) for storing data by a systematic dissemination through the integrated sensor nodes. It guarantees that information could be retrieved in each piece of the concrete by intelligently managing data replication among each neighborhood of the sensor network. The protocol considers in the same set uniformity storage in the whole network, the data importance level, and the resource constraints of sensor nodes. Then, another new data retrieval protocol (RaWPG) is developed to extract the stored information. Castalia/OMNeT++ simulator is used to evaluate the performances of the proposed protocols.

Keywords

Internet of Things, product lifecycle management, communicating material, wireless sensors network.