



LE BOUNDED MESSAGE PROTOCOL :

UN PROTOCOLE DE
COMMUNICATION BASÉ SUR LA
DISTANCE ENTRE LES OBJETS
DE L'INTERNET DES OBJETS

PAR FRÉDÉRIC BERGERON



Catalogage avant publication de Bibliothèque et Archives nationales du Québec et Bibliothèque et Archives Canada

Titre: Le bounded message protocol : un protocole de communication basé sur la distance entre les objets de l'internet des objets / Frédéric Bergeron.

Noms: Bergeron, Frédéric, 1991- auteur.

Description: Comprend des références bibliographiques. | Textes en français et en anglais.

Identifiants: Canadiana 20240010205F | ISBN 9782981901743 (PDF)

Vedettes-matière: RVM: Internet des objets. | RVM: Protocoles de réseaux d'ordinateurs. | RVM: Systèmes de géolocalisation en intérieur.

Classification: LCC TK5105.8857.B47 2024 | CDD 004.67/8—dc23

TABLE DES MATIÈRES

<i>Liste des figures</i> -----	7
<i>Liste des tableaux</i> -----	9
<i>Abréviations</i> -----	11
<i>Remerciements</i> -----	16
INTRODUCTION -----	19
<i>PREMIÈRE PARTIE - PRÉLIMINAIRES</i>	24
Chapitre 1 : Objectifs et méthodologie -----	26
1.1 Objectifs -----	27
1.2 Hypothèses -----	29
1.3 Méthodologie -----	29
1.3.1 Méthodologie pour le BMP -----	30
1.3.2 Méthodologie pour IPADL -----	32
Chapitre 2 : État de l'art -----	34
2.1 L'Internet des objets -----	35
2.1.1 IdO en action -----	36
2.1.2 La pérennité de l'IdO -----	42
2.2 Sur les protocoles de communication -----	46
2.2.1 Les protocoles généraux -----	47
2.2.2 Les sans-fils -----	50
2.3 Sur les technologies de positionnement -----	57
2.3.1 Positionnement extérieur -----	57
2.3.2 Positionnement intérieur -----	58

<i>DEUXIÈME PARTIE - CONTRIBUTIONS EN COMMUNICATION</i>	060
Chapitre 3 : Proposition du BMP -----	67
3.1 Modèle de propagation -----	70
3.2 Réseau filaire -----	73
3.3 Portée des données -----	73
3.4 Modulation de la puissance d'émission des capteurs-----	79
3.5 Gestion du temps -----	80
3.6 Entreposage des données -----	82
3.6.1 Éléments mobiles -----	83
3.6.2 Temps et entreposage -----	85
3.7 Format des messages -----	86
3.8 Retour sur les objectifs -----	91
Chapitre 4 : Implémentation -----	94
4.1 Caractéristiques communes -----	97
4.1.1 Étapes de traitement -----	97
4.1.2 Représentation interne -----	103
4.1.3 Gestion des collisions -----	104
4.2 Différences entre les implémentations -----	105
4.3 Retour sur les objectif -----	108
Chapitre 5 : Performance du BMP -----	109
5.1 Plateforme d'essai de Blagnac -----	110
5.2 Évaluations -----	113
5.3 Appartement intelligent du DOMUS -----	117
5.3.1 Scénarios -----	120
5.3.2 Résultats et discussions -----	123
5.4 Retour sur les objectifs -----	128

Chapitre 6 : Utilisations et travaux futurs -----	130
6.1 Aspects pratiques -----	131
6.1.1 Conditions prévues d'utilisation -----	132
6.1.2 Applications cibles -----	133
6.1.3 Limitations et garanties du protocole -----	134
6.2 Cas d'usage prévus -----	136
6.2.1 Assistance à l'errance nocturne -----	137
6.2.2 Intersection intelligente -----	139
6.3 Conclusion -----	146
 <i>TROISIÈME PARTIE - CONTRIBUTIONS EN POSITIONNEMENT</i>	 147
 Chapitre 7 : Tracking Objects Within a Smart Home -----	 152
7.1 Introduction -----	157
7.2 Related work -----	163
7.3 Static positioning inside a smart home -----	166
7.3.1 Smart home -----	167
7.3.2 Special object -----	171
7.3.3 Logical zones -----	171
7.3.4 Data collection -----	173
7.4 Building models for the IPS -----	174
7.4.1 Multilayer perceptron -----	175
7.4.2 K-nearest-neighbours -----	176
7.4.3 Bayesian network -----	177
7.4.4 Naive Bayes -----	178
7.4.5 Classic trees -----	178
7.4.6 Trees mixed with other models -----	180
7.4.7 Trees forest -----	182
7.4.8 Discussion -----	182
7.5 Dynamic tracking system -----	183
7.5.1 Filters -----	185

Table des matières

7.6	Experiments -----	191
	7.6.1 Tracking data -----	192
	7.6.2 Evaluation metrics -----	194
	7.6.3 Results -----	195
	7.6.4 Execution time -----	199
7.7	Discussion -----	201
	7.7.1 Effect of speed -----	202
7.8	Conclusion -----	203
	7.8.1 Contributions -----	204
7.9	Acknowledgements -----	205
Chapitre 8 : RFID Indoor Localization Using Statistical Features --		206
8.1	Related work -----	213
8.2	Methodology -----	215
	8.2.1 Smart home -----	216
	8.2.2 Dataset -----	218
	8.2.3 Statistical features -----	219
8.3	Experiments and results -----	221
	8.3.1 Raw readings classification -----	221
	8.3.2 Statistical features classification -----	223
	8.3.3 The impact of windowing -----	224
	8.3.4 Selection of features -----	226
8.4	Conclusion -----	230
8.5	Acknowledgements -----	232
CONCLUSION-----		234
BIBLIOGRAPHIE-----		243

LISTE DES FIGURES

Figure 2.1 :	Définition de l'Internet des objets	36
Figure 3.1 :	Exemple d'une route intelligente	77
Figure 3.2 :	Exemple d'application à la marine	78
Figure 3.3 :	En-tête du protocole	87
Figure 4.1 :	Étapes de traitement d'un message	99
Figure 4.2 :	Étapes de l'envoi d'un message	102
Figure 5.1 :	Plan de la plateforme d'essai de l'IUT de Blagnac	111
Figure 5.2 :	Position des objets dans le DOMUS et trajet de l'utilisateur	119
Figure 6.1 :	Description d'une intersection	144
Figure 7.1 :	Reference tag method	164
Figure 7.2 :	A picture of the DOMUS smart home	168
Figure 7.3 :	A map of the DOMUS smart home	169
Figure 7.4 :	A picture of the custom object	170
Figure 7.5 :	Accuracy given K in k-nearest neighbours	176

Table des matières

Figure 7.6 :	Data flow inside the ITS	184
Figure 7.7 :	Map of the kitchen with raw data versus the limiting filter	188
Figure 7.8 :	Example of the filters	189
Figure 7.9 :	Experimental set-up	193
Figure 7.10 :	Results for the targeted zones found metric	196
Figure 7.11 :	Results for the sequential targeted zones found metric	197
Figure 8.1 :	Map of the smart home and position of the RFID antennas	217
Figure 8.2 :	Accuracy of the J48 classifier	225

LISTE DES TABLEAUX

Tableau 3.1 :	Modes de positionnement	75
Tableau 5.1 :	Résultats de la première itération du scénario	124
Tableau 5.2 :	Résultats combinés des itérations 2 à 9 du scénario	128
Tableau 6.1 :	Exemples d'utilisation du BMP	138
Table 7.1 :	Comparison of radio frequency technologies for tracking	159
Table 7.2 :	Accuracy and distances with non tree models	175
Table 7.3 :	Accuracy and distances with the trees	179
Table 7.4 :	Weighting function for the pre-filter	187
Table 7.5 :	Effect of N and the weight function with the moving average in the kitchen	198
Table 7.6 :	Limiting filter with the moving average	199
Table 7.7 :	Execution time of the ITS	200
Table 8.1 :	The list of features applied on the RFID readings	220
Table 8.2 :	Accuracy of some classification algorithms on the raw dataset	222

Table des matières

Table 8.3 :	Accuracy on the feature dataset with a window of 5	223
Table 8.4 :	Accuracy of J48 when removing features with window of 5	228
Table 8.5 :	Individual features with window of 5	229

ABRÉVIATIONS

AES :	Advanced Encryption Standard
ARP :	Address Resolution Protocol
ASCII :	American Standard Code for Information Interchange
BLE :	Bluetooth Low Energy
BMP :	Bounded Message Protocol
CRTC :	Canadian Radio-television and Telecommunications Commission
CSMA/CA :	Carrier Sense Multiple Access with Collision Avoidance
CTS :	Clear to Send
DNS :	Domain Name Server
HTTP :	HyperText Transfer Protocol
IdO :	Internet des objets
IP :	Internet Protocol
IPADL :	Indoor Positioning for Activities of Daily Living
IPS :	Indoor Positioning System

IPv4 :	Internet Protocol version 4
IPv6 :	Internet Protocol version 6
ITS :	Indoor Tracking System
IUT :	Institut universitaire de technologie
LAN :	Local Area Network
LTE :	Long-Term Evolution
MAC :	Media Access Control
MMS :	Multimedia Message Service
MQTT :	Message Queuing Telemetry Transport
NFC :	Near Field Communication
OSI :	Open Systems Interconnection
PAN :	Personal Area Network
PCA :	Principal Components Analysis
RFID :	Radio Frequency Identification

ROM :	Read-Only Memory
RSSI :	Received Signal Strength Indication
RTS :	Request to Send
SMS :	Short Message Service
SSID :	Service Set IDentifier
STZF :	Sequential Targeted Zones Found
TCP :	Transmission Control Protocol
TTL :	Time to Live
TZF :	Targeted Zones Found
UCT :	Unité centrale de traitement
UDP :	User Datagram Protocol
UTC :	Coordinated Universal Time
UWB :	Ultra-Wide Band
VANET :	Vehicular Ad hoc Network

Table des matières

WAN : Wide Area Network

WLAN : Wireless Local Area Network

« Never pass up a good thing. »

— Jean Raszak, *Starship Troopers*, Paul Verhoeven, 1997

REMERCIEMENTS

Par cette thèse, j'exauce le vœu de ma grand-mère. Je suis docteur. Malheureusement, je ne pourrai pas te soigner, mais je pourrai concevoir la technologie qui pourra t'aider. Ce parcours fut riche, mais surtout éducatif.

À Béatrice, qui aura su me rappeler qu'en 18 mois de vie on apprend nettement plus qu'en 3 ans de thèse. Ta vie n'est encore qu'une vague sur l'océan de la mienne et je peux déjà dire que cette vague va marquer plus d'un littoral. Puissent nos premiers moments de communion n'être que les premiers d'une série innombrable.

À Josh, qui du haut de ses trois semaines m'enseigne que baver n'est que le résultat visible quand le résultat important est le renforcement intérieur. Si Béatrice était attendue pour 2019, tu es de loin la plus belle surprise de 2020.

À Yanick et Andréanne, à Jessica et Mickaël, votre confiance me transperce au plus profond de mon être, jamais je ne pourrai exprimer totalement la reconnaissance que je vous porte.

À Wathek, Hubert, Mathieu, Armel et tous les autres du Domus que l'espace m'empêche de nommer. Vous m'avez permis de m'accomplir, de me dépasser, mais surtout de réfléchir sur ma place dans le monde, et je l'espère, dans votre monde. Vous avez su m'apporter beaucoup plus qu'aucune thèse ne pourrait le faire, prouvant une fois de plus qu'en l'humain réside l'espoir des jours lumineux.

À Kevin, qui le premier aura su voir en moi un candidat à la maîtrise. À Sylvain, qui aura su encourager mes idées, et à l'occasion les « challenger » pour voir ce qu'il y a plus loin. À Sébastien, pour sa force tranquille et sa curiosité sans bornes pour mon travail. Vous trois aurez été mes directeurs, ceux qui auront stimulé cet accomplissement supérieur à mes attentes.

À mes parents, Manon, Mario, Lucielle, Louis-Philippe et Florence. Vos encouragements continus m'auront soutenu tout au long de cette épreuve. À André, ta présence a été bien sentie à de nombreuses reprises. À Ruth, Joëlle et Geneviève, vous aurez su me rappeler à quel point cette thèse n'est pas triviale, accentuant ainsi son impact et ma motivation.

À ces dizaines, ces centaines, ces milliers de gens fabuleux que j'ai croisés au cours de ces 10 ans. Votre apport ne saurait être noté avec justesse, malgré son caractère essentiel. À vous tous que j'ai nommés, à vous tous que je ne saurais identifier, vous êtes l'équation de ma solution. Cette thèse est pour vous, puissiez-vous l'endosser au même titre que vous m'avez fait confiance toutes ces années.

« Revenu annuel : vingt livres ; dépenses annuelles : dix-neuf livres, dix-neuf shillings, six pence ; résultat : le bonheur.

Revenu annuel : vingt livres ; dépenses annuelles : vingt livres, zéro shilling, six pence ; résultat : la misère. »

— Charles Dickens, *David Copperfield*, 1850

INTRODUCTION

L'introduction d'Internet chez les particuliers a créé de nombreuses occasions de découvertes. Les possibilités de communications ont vite été décuplées et les ordinateurs personnels sont devenus d'autant plus intéressants que le matériel est devenu plus abordable. Les fabricants ont poursuivi leur course à la miniaturisation avec encore plus d'ardeur et bientôt les téléphones intelligents ont vu le jour. Peu après, en coulisse, les chercheurs en informatique ont vu qu'il y avait un potentiel d'inclusion de matériel informatique dans les objets de la vie courante. Le rêve de l'informatique ubiquitaire n'était pas nouveau, mais il était maintenant possible. Il était désormais envisageable de permettre à des objets anodins de communiquer entre eux. D'abord filaires, ces communications sont devenues sans fil à mesure que le marché maintenant traditionnel des téléphones portables a stimulé la création de nouveaux protocoles plus performants. Ainsi, Bluetooth [142], connu du grand public comme étant une façon d'échanger des contacts et de la musique avec des téléphones à proximité, permet aussi des échanges d'informations entre différents objets. Les objets dans lesquels on a inclus des processeurs informatiques sont maintenant nommés des **objets intelligents**. Les objets intelligents font plus généralement référence aux objets capables de collecter de l'information sur l'environnement à l'aide de capteurs, aux objets capables d'agir sur l'environnement à l'aide d'effecteurs et dans une certaine mesure aux objets capables de servir d'intermédiaire entre les capteurs et les effecteurs. Ces objets communiquent entre eux sur des réseaux informatiques, filaires ou sans fil. La réunion de ces réseaux forme alors un Internet, voire une extension de l'Internet grand public. Cette réunion de réseaux forme un **Internet des objets** (IdO), bien que le terme « IdO » soit souvent employé pour désigner tout réseau d'objets par abus de langage. Cette thèse utilisera le terme « IdO » pour désigner tout réseau d'objets [91].

Contexte

L'introduction des objets intelligents dans de nombreux aspects de la vie a décuplé les capacités de collecte d'information à propos de l'environnement dans lequel nous évoluons. On retrouve des objets intelligents partout, au point qu'il n'est maintenant plus possible de faire une liste d'endroits. Les maisons intelligentes, l'industrie 4.0 [109] et les voitures intelligentes sont quelques exemples de termes maintenant familiers indiquant que les objets intelligents ont modifié les façons de faire de nombreux domaines.

L'omniprésence des capteurs et des effecteurs ouvre la porte à une automatisation de nombreux processus. Elle invite aussi à la collecte d'un maximum de données dans le but d'améliorer ces processus, que ce soit pour les rendre plus performants ou simplement plus pertinents. Les usages sont aussi multiples que variés. D'un côté, l'industriel a avantage à connaître un maximum d'information sur sa chaîne de montage afin de prévenir les accidents. D'un autre côté, un ergothérapeute pourra fournir une évaluation moins biaisée d'une personne évoluant seule dans son environnement rendu intelligent que dans un environnement inconnu dans un centre d'évaluation [144, 120]. D'un troisième côté, une ville souhaitera suivre en temps réel la circulation routière pour détecter les points de frictions et proposer des solutions. Les applications de l'IdO sont infinies. Les capacités des réseaux actuels ainsi que les ressources disponibles sont toutefois finies et certaines limites commencent à apparaître.

Problématique

Chaque nouveau capteur apporte de nouvelles données. Chaque nouvel effecteur apporte de nouveaux moyens d'action. Ensemble, ils apportent toujours plus de fonctionnalités. Il en résulte une pression sur les infrastructures de calcul et de communication. En effet, il ne suffit pas de collecter des données sur un capteur, il faut aussi les traiter pour en extraire de l'information utile. Souvent, les données doivent être modifiées pour devenir utiles. Un capteur de température, par exemple, risque de renseigner sur une tension en volt. Il faut alors utiliser une table de conversion pour extraire une température. Un algorithme voulant utiliser toutes les données d'un environnement doit aussi être capable de sélectionner celles qui sont pertinentes pour chaque tâche. Un capteur peut aussi effectuer des mesures à une fréquence plus élevée que celle possible sur le serveur de calcul. Aussi, le serveur de calcul peut devenir tellement engorgé qu'il n'arrive plus à suivre le rythme qu'il avait lui-même établi et commence alors à prendre du retard et à ne plus fonctionner en temps réel. Le problème du manque de puissance de calcul est d'autant plus présent dans une architecture centralisée où toutes les données de tous les capteurs sont rassemblées vers un point central qui doit les traiter pour ensuite envoyer des commandes aux effecteurs et fournir des rapports sur les données. Il a d'ailleurs été projeté en 2014 qu'une ville intelligente puisse produire jusqu'à 4,1 téraoctets de données par jour par kilomètre carré dès 2016 [54].

De nombreux réseaux informatiques ont été conçus avant l'avènement des objets intelligents. Ces objets viennent alors s'ajouter aux postes informatiques traditionnels pour avoir leur part de bande passante. Avec plus d'utilisateurs, le réseau devient plus chargé et peut atteindre le point de saturation plus rapidement. Il n'est pas toujours facile d'augmenter la capacité d'un réseau, ce qui devient alors un problème.

Introduction

Au-dessus du médium de communication, les fabricants d'objets possèdent souvent leur propre protocole de communication, parfois propriétaire. Il devient alors difficile de faire communiquer entre eux des objets provenant de différents manufacturiers. Certaines alliances ont conduit à des protocoles plus généraux, tels Bluetooth, Zigbee et KNX, mais aucun consensus sur un protocole unique ne semble sur le point d'émerger. Chaque protocole est conçu pour tirer avantage des particularités de la technologie des fabricants, rendant une unification des protocoles peu probable sans perte de fonctionnalités pour quiconque.

Au-delà des limitations matérielles, il y a aussi des problématiques conceptuelles aux façons centralisées de considérer l'IdO. Souvent, il n'est pas nécessaire de rassembler toute l'information à un seul endroit. Les récents scandales de vols de données dans les grandes compagnies [77] nous enseignent qu'il existe aussi un risque important de sécurité à rassembler les données. Que ce soit pour un enjeu de sécurité ou d'efficacité, il devient alors intéressant de conserver localement les données nécessaires à une tâche. De même, il peut être intéressant de détruire des données après un certain temps. Les données de capteur, particulièrement, sont continuellement mises à jour. On peut alors se questionner sur la pertinence de conserver chaque donnée plutôt que de conserver uniquement les k dernières lectures. Les protocoles actuels de communication ne permettent pas de mettre en place facilement ces éléments.

Cette thèse propose une nouvelle façon de concevoir les systèmes d'IdO qui déplace l'intelligence du serveur central vers les objets eux-mêmes. Ce changement est imposé par un nouveau protocole de communication basé sur la distance entre les objets. Le temps de vie des messages est aussi pris en considération directement par le protocole, enlevant cette responsabilité à l'application utilisant le protocole.

Structure de la thèse

Les travaux présentés dans cette thèse s'échelonnent sur près de cinq ans et sur deux cycles universitaires, à savoir la maîtrise et le doctorat. Pour bien représenter cette situation et l'évolution de l'objectif des travaux dans le temps, ce document sera divisé en trois parties. Une première partie présentera l'état de l'art nécessaire à la bonne compréhension des nombreux sujets abordés dans les parties subséquentes. L'état de l'art traitera plus spécifiquement des technologies de positionnement, de l'Internet des objets et finalement des protocoles de communication.

La seconde partie présente la contribution principale de cette thèse, à savoir un protocole de communication pour l'IdO basé sur la distance entre les objets. Ce protocole se nomme le BMP, pour *Bounded Message Protocol*. La partie débute par une présentation de la genèse du protocole. Suivent les descriptions théorique et fonctionnelle du protocole puis une étude de ses performances dans un environnement contrôlé. La section se termine par une présentation de certains éléments ciblés concernant les futurs développements du protocole et sur certaines applications potentielles.

La troisième partie se concentre sur la notion de position. Cette notion est centrale au protocole développé dans la seconde partie. Les contributions présentées dans cette partie se concentrent autour d'un système de positionnement intérieur d'objets et de moyens d'améliorations de ce système. Cette partie prend la forme de deux articles de journaux scientifiques révisés par les pairs.

PREMIÈRE PARTIE

PRÉLIMINAIRES

« May you live as long as you wish and love as long as you live. »

— Robert A. Heinlein, *Time Enough for Love, Ace*, 1973



PREMIER CHAPITRE

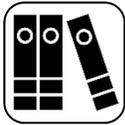
Objectifs et méthodologie

1.1 Objectifs



1.2 Hypothèses

1.3 Méthodologie



Dans le chapitre précédent, l'Internet des objets et certains de ses défis ont été introduits. Ce chapitre va préciser de quelle façon ces défis seront abordés dans le cadre de cette thèse. Cette précision se fera tout d'abord par la présentation des objectifs spécifiques à ce travail, puis des hypothèses devant être posées. Finalement, une présentation de la méthodologie conclura ce chapitre.

1.1 Objectifs

L'introduction a présenté l'IdO comme étant l'ensemble des réseaux permettant aux objets intelligents de communiquer à la fois entre eux et avec l'Internet au sens conventionnel [10]. Bien souvent, les protocoles permettant de communiquer entre objets ne sont pas les mêmes que ceux donnant accès à Internet et il est alors nécessaire d'avoir un objet capable de faire le pont entre les deux types de réseau. Une couverture plus vaste de ces protocoles réseau sera offerte dans le chapitre 2 traitant de l'état de l'art.

Une tendance actuelle dans l'IdO est de communiquer l'information de terrain recueillie par les différents objets vers un serveur qui va analyser ces données et renvoyer les éventuelles actions vers les objets concernés [4]. Cette méthode ne s'applique pas à toutes les situations. Il y a certains cas où les données devraient rester locales, pour des questions de sécurité par exemple. Dans d'autres cas, l'ensemble des données nécessaires à une tâche peut être contenu dans l'environnement immédiat et le transit vers le serveur central n'est pas réellement requis.

Chapitre 1 – Objectifs et méthodologie

Cette thèse vise à proposer une méthode décentralisée pour distribuer la prise de décision dans un système d'IdO. Cette méthode prend la forme d'un protocole de communication conçu de façon à faciliter cette décentralisation. Le protocole se veut aussi en accord avec les principes de l'informatique autonome [87], du moins dans les aspects pouvant être de sa responsabilité. Décentraliser les décisions est un défi de l'IdO qui fait écho à un autre défi, celui de la quantité des données. Filtrer l'important pour une tâche est parfois difficile pour un serveur central [146, 83]. Un serveur distribué aura globalement peut-être plus de données à traiter vu la redondance possible entre les objets, mais chaque objet devrait avoir moins d'information à analyser qu'un serveur central.

Plus généralement, les objectifs sont de :

1. Concevoir un protocole de communication basé sur les positions des objets;
2. Concevoir un protocole de communication capable de gérer une durée de vie des messages;
3. Concevoir un système de positionnement intérieur rapide pour objets;
4. Identifier et vérifier les limitations du protocole conçu;
5. Déployer le protocole dans un scénario se rapprochant des conditions réelles.

Le protocole présenté dans ce document utilise un mode de réplication des messages basé sur la position des objets. Ce protocole a pour nom BMP. Les premiers objectifs sont naturellement de concevoir le protocole en question. Le troisième concerne le besoin primordial au BMP d'offrir une méthode de positionnement fiable des objets dans un environnement intérieur. Les derniers objectifs concernent la vérification du fonctionnement du protocole.

1.2 Hypothèses

Ce travail est basé sur un certain nombre d'hypothèses. Une première série d'hypothèses est en lien direct avec le protocole BMP :

1. Il est possible de concevoir un protocole de communication n'utilisant pas d'adresse de destination;
2. Il n'est pas nécessaire de faire partie d'un réseau structuré pour communiquer;
3. Tout objet est capable de mesurer l'écoulement du temps avec régularité;
4. Il est possible de concevoir des scénarios d'assistance technologique à l'autonomie décentralisée à partir d'informations partielles sur l'environnement.

Une seconde série traite plus particulièrement de la problématique du positionnement :

5. Il est possible pour tout objet d'obtenir sa position à la précision nécessaire à son travail en un temps raisonnable à ce travail;
6. La RFID passive est une technologie permettant de faire un positionnement intérieur fiable;
7. Un système d'intelligence artificielle peut résoudre le problème du positionnement intérieur.

1.3 Méthodologie

Tout comme cette thèse, la méthodologie se découpe en trois parties. La première, celle-ci, donne les bases du travail. La deuxième partie traite du protocole BMP et la dernière s'attarde plus directement à la question du positionnement intérieur.

1.3.1 Méthodologie pour le BMP

Dès le début, ce travail est motivé par l'idée qu'il est possible d'élaborer un protocole de communication n'utilisant pas le système habituel des adresses de destination pour les messages. Le premier objectif de la thèse reprend cette idée.

La première étape pour atteindre cet objectif est d'analyser les protocoles actuels pour évaluer dans quelle mesure le système envisagé existe déjà. Il en ressort que les limitations des protocoles existants justifient la création d'un nouveau protocole. Une liste des besoins est alors établie et certains sont choisis pour la conception du BMP. Le protocole doit être décentralisé, doit conserver localement ses informations et doit offrir un mécanisme d'expiration des données.

Dans tout processus de conception, il est nécessaire d'effectuer certains choix. Ces choix découlent parfois de résultats antérieurs et peuvent alors s'établir de manière empirique. Ces choix seront présentés dans les chapitres suivants. D'autres choix ne reposent sur aucune preuve a priori et doivent donc s'appuyer sur un certain nombre d'hypothèses qui pourront ensuite être confirmées ou infirmées. Le délai avant la retransmission des messages du BMP et la taille des zones de l'algorithme de positionnement sont les principaux choix empiriques de cette thèse. L'algorithme de positionnement décrit en détail au chapitre 7 fonctionne en divisant l'espace en zones, un peu à la manière d'un jeu d'échecs.

La vérification du premier objectif s'appuie sur certaines hypothèses. La première, nécessaire à toute chose, est qu'il est possible de le faire. La seconde hypothèse est qu'il n'est pas nécessaire de faire partie d'un réseau pour communiquer. Cela signifie qu'il n'est nullement nécessaire de savoir à qui on s'adresse, ni même si un autre objet écoute au moment de communiquer. Les humains ont l'habitude de ce genre de communication (les arts en sont un exemple), mais ce n'est pas le cas en informatique. L'hypothèse 2 sous-tend donc qu'il peut être intéressant de le faire. Une formulation équivalente de cette hypothèse est qu'il suffit d'avoir accès à un médium de communication pour communiquer.

Le second objectif provient d'un besoin secondaire identifié lors de la réalisation du premier objectif. L'objectif 1 permet la mise en place d'un protocole permettant de limiter la propagation d'un message à un rayon mesurable donné. Le second objectif ajoute une couche limitative en venant aussi préciser une durée de vie aux messages. Ensemble, ces deux objectifs demandent à ce qu'une information ne puisse vivre que dans un espace donné pour un temps donné. L'objectif 2 n'est toutefois possible que si l'hypothèse 3 est valide, à savoir que tous les objets sont capables de mesurer le passage du temps avec régularité, c'est-à-dire que l'écart entre le temps mesuré et le temps réel écoulé est négligeable par rapport au temps de vie d'une donnée.

La méthodologie pour atteindre ces deux premiers objectifs est théorique. Il s'agit de réfléchir sur les problèmes soulevés par la revue de la littérature pour proposer une résolution théorique. Les chapitres dédiés à ces objectifs présenteront surtout le produit fini plutôt que l'évolution de la réflexion s'associant au BMP.

Le quatrième objectif vise à vérifier que les procédés de contrôle du protocole fonctionnent. Cette vérification est principalement empirique. Pour y parvenir, le candidat a eu accès à la plateforme d'essais de l'IUT de Blagnac, en France [152]. Différents essais ont pu y être menés suivant les différentes caractéristiques du protocole. Tout d'abord, le mode de propagation des messages dans un mode multi-sauts a été vérifié. Puis, ce fut le tour du mécanisme de gestion du temps et ensuite du mécanisme de gestion de la distance. Finalement, il a été possible d'évaluer la résistance à la montée de charge dans des conditions s'approchant de celles présentes dans une maison intelligente pour l'évaluation de l'objectif 5. La plateforme d'essais de l'IUT de Blagnac ayant déjà maintes fois été éprouvée avec succès pour la création et la vérification de protocoles de communication [153, 52], il n'est pas posé en hypothèse qu'elle puisse le faire. Le cinquième objectif est exploré une seconde fois par un déploiement d'objets au sein du laboratoire DOMUS dans une configuration réaliste. Les objets sont placés aux endroits où des objets intelligents devraient se situer (devant le grille-pain, par exemple) et l'expérience consiste à les faire communiquer de façon réaliste.

1.3.2 Méthodologie pour le IPADL

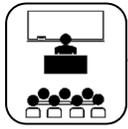
La méthodologie pour le système de positionnement est un peu plus directe. L'objectif 3 régit l'entièreté de cette partie en nécessitant une méthode de positionnement fonctionnant à l'intérieur qui sera rapide d'exécution. Cette partie du projet utilise la technologie RFID passive [38] combinée avec de l'intelligence artificielle pour créer un système de positionnement intérieur. Les hypothèses 6 et 7 posent donc que ces choix sont valides.

Pour répondre à l'objectif, il a été choisi de considérer le problème de la position comme étant un problème de classification au lieu d'un problème de régression comme c'est souvent le cas dans la littérature. Cette reformulation donne accès à un vaste éventail de méthodes reconnues disposant d'une implémentation dans l'outil Weka [56]. La première étape a alors été de diviser l'appartement du DOMUS en une série de zones qu'il serait possible de classifier et puis de collecter des valeurs RSSI (*Received Signal Strength Indication*) provenant d'un objet, pour chacune de ces zones. La taille des zones a été décidée empiriquement, de manière à fournir un équilibre entre la taille des zones et la qualité du positionnement. Après, il s'agit d'entraîner différents modèles et de choisir le meilleur selon les critères de performance et de temps d'exécution. Le choix des modèles a été arbitraire, avec une préférence pour les arbres en raison de leur rapidité d'exécution. Le système résultant se nomme IPADL, pour *Indoor Positioning for Activities of Daily Living*.

Une seconde réflexion sur la méthode a permis d'établir que certaines mesures statistiques sur les données pouvaient permettre d'augmenter la précision du système pour un coût en temps modique.

Ce travail de positionnement peut être considéré comme indépendant du BMP. Toutefois, les objets du BMP ont besoin de pouvoir obtenir leur position et IPADL est un système pouvant la leur donner.

L'hypothèse 5 stipule que dans le cadre du BMP les objets doivent avoir accès à une donnée de position fiable et rapide. L'exécution de l'objectif 3 permet de vérifier qu'IPADL répond à ces critères. Aussi, il n'y a a priori aucune raison pour qu'IPADL ne puisse pas fonctionner à l'extérieur, dans la limite de portée des antennes. Cela n'a toutefois pas été vérifié.



DEUXIÈME CHAPITRE

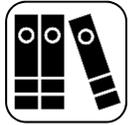
État de l'art



2.1 L'Internet des objets

2.2 Sur les protocoles de communication

2.3 Sur les technologies de positionnement



Cette thèse touche à de nombreux aspects de l’informatique. Ce chapitre va présenter l’état de l’art de ces différents aspects. Cette démarche suit deux buts bien distincts : premièrement, celui de fournir les bases sur lesquelles les prochains chapitres sont construits et, deuxièmement, celui de permettre de mieux faire ressortir les contributions de cette thèse. Le chapitre est divisé en trois thèmes :

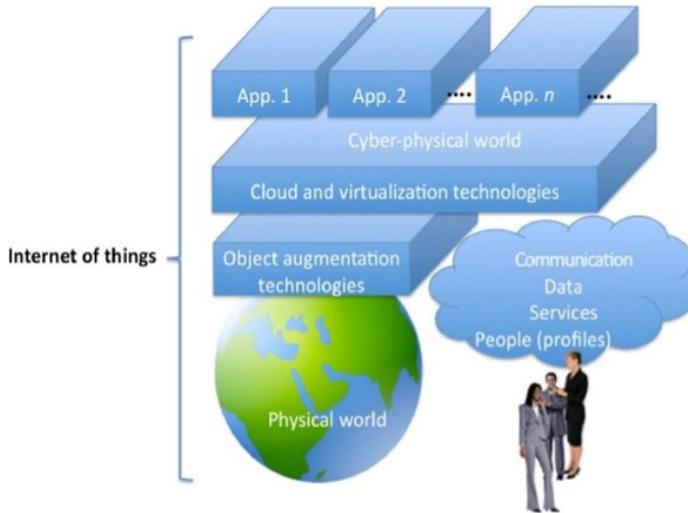
- L’Internet des objets. Ce thème explore les défis de l’IdO et les concepts d’informatique autonome.
- Les protocoles de communication. Ce thème explore certains protocoles dont les caractéristiques ont inspiré le BMP.
- Le positionnement d’objets. Ce dernier thème s’attarde à la notion de position et aux techniques permettant de l’acquérir. Ce thème sera un peu moins exploré que les autres puisqu’il revient dans les articles issus de ces travaux, aux chapitres 7 et 8.

2.1 L’Internet des objets

Le paradigme de l’Internet des objets, au sens large, fait référence à un ensemble d’objets connectés œuvrant ensemble pour l’atteinte d’un objectif commun. Une définition universelle de l’IdO n’existe pas vraiment puisqu’elle diffère selon le point de vue considéré. Les différentes visions sont orientées soit sur les objets, soit sur Internet ou encore sur la sémantique, comme l’ont expliqué Atzori et al. dans l’une des premières revues sur le sujet [4]. Dans un second article sur le sujet [5], les mêmes auteurs apportent une mise à jour sur leur définition pour y inclure davantage d’éléments. Selon eux, l’Internet des objets inclut les domaines de la conception des objets, de leur utilisation, de la communication, voire jusqu’à l’intelligence artificielle derrière leur utilisation. Leur vision est résumée à la figure 2.1, extraite de leur article. La suite de cette section sur l’Internet des objets sera divisée en deux parties. Une première partie va présenter des utilisations de l’IdO et une seconde insistera sur les défis actuels et à venir.

Figure 2.1

Définition de l’Internet des objets selon [5].



2.1.1 IdO en action

Les objets connectés sont présents dans pratiquement tous les aspects de la vie. En dresser une liste exhaustive n’est donc pas envisageable. De plus, de nouveaux objets connectés arrivent sur le marché continuellement. Cette partie va plutôt se concentrer sur certains domaines d’application typiques, sans entrer trop dans les détails des objets impliqués ni des algorithmes exacts utilisés pour chaque application. Dans de très nombreux cas, les applications présentées ne sont pas limitées à l’usage exclusif d’un seul domaine.

Les maisons intelligentes

Le premier domaine qui vient en tête lorsqu’on évoque l’IdO est habituellement celui de la maison intelligente (*smart home*). Ce thème est présent depuis longtemps dans les œuvres de science-fiction et fait partie de l’imaginaire collectif depuis bien avant sa mise en application réelle. Bien que la réalité n’ait pas encore dépassé la fiction, l’écart se resserre à un rythme accéléré depuis quelques années [81].

Les maisons intelligentes peuvent être considérées du point de vue commercial, du point de vue médical ou encore du point de vue de l’assistance technologique. Le point de vue commercial cherche principalement à automatiser certaines tâches ou à rassembler et simplifier les interfaces de contrôle des objets. C’est dans cette catégorie qu’il convient de classer les enceintes connectées de Google (Google Nest), Amazon (Echo/Alexa) et Apple (HomeKit). À travers ces différentes technologies, il est possible d’interagir avec une foule d’objets vendus séparément, par ces mêmes compagnies ou par des tiers. Ces enceintes ont la particularité d’être contrôlées par la voix. La configuration peut se faire à partir d’un téléphone mobile ou d’un ordinateur, mais l’usage est habituellement vocal. L’usager dit ses commandes à haute voix et l’enceinte les exécute. Il est aussi possible de définir des routines pour qu’un ensemble de tâches s’exécute à la suite d’un certain stimulus. Dans le même ordre d’idée, le service If This Then That (ITTT) permet de connecter de nombreux objets pour ensuite définir un ensemble de règles pour contrôler ces objets afin de faciliter les interactions.

Dans la catégorie des outils grand public, mais sans pour autant être commerciaux, il y a la plateforme libre openHAB⁵ [89] qui permet d’obtenir le même genre de contrôle sur les objets domotiques de la maison. openHAB a toutefois la particularité de pouvoir être installée sur n’importe quel système d’exploitation. Sa flexibilité s’étend aussi à de très nombreuses technologies et à sa capacité de s’intégrer avec les plateformes commerciales mentionnées précédemment.

Un avantage d’openHAB est qu’elle peut fonctionner avec ou sans accès à l’infonuagique, donnant un plus grand contrôle à l’utilisateur sur qui a accès aux données. Des points de vue de l’assistance à domicile et de l’assistance médicale, les maisons intelligentes offrent déjà une très grande variété de services, de nouveaux étant publiés dans la littérature sur une base régulière. La présence de nombreux capteurs et actionneurs dans un environnement permet d’inférer le contexte de déroulement des activités quotidiennes. Le contexte, jumelé avec les mêmes données brutes, permet ensuite de reconnaître les activités, les états d’esprit, les postures et les situations dangereuses vécues par les occupants dudit environnement. Dans un effort collectif, les membres du laboratoire DOMUS ont rédigé un chapitre de livre résumant les modèles et les services permettant d’atteindre ces objectifs de reconnaissance et d’assistance [3].

Les villes intelligentes

Alors que la recherche sur les maisons intelligentes s’intéresse principalement au confort, à la sécurité et à l’assistance médicale des occupants, la recherche sur les villes intelligentes (*smart city*) porte généralement sur la découverte de services et l’optimisation des infrastructures locales [163]. Selon Kim et al., les trois composantes requises pour former une ville intelligente sont d’avoir une ville instrumentée, interconnectée et intelligente [78].

⁵ <https://www.openhab.org>

C’est seulement si ces trois composantes sont présentes qu’une ville peut alors être qualifiée d’intelligente, selon ces auteurs. Forrester [160] propose de son côté la définition suivante : « *The use of Smart Computing technologies to make the critical infrastructure components and services of a city — which include city administration, education, healthcare, public safety, real estate, transportation, and utilities — more intelligent, interconnected, and efficient.* »

Pour Forrester, il s’agit donc d’utiliser l’IdO pour rendre les infrastructures critiques et les services d’une ville plus intelligents, interconnectés et efficaces.

Un secteur important de la recherche sur les villes intelligentes est celui de la communication. Lorsqu’il s’agit de collecter et de traiter les informations de toute une ville, la question de leur transport de façon efficace devient vite importante. Il y a deux aspects à cette question. Un premier aspect concerne le protocole à utiliser et le second concerne l’architecture matérielle et logicielle supportant ce protocole.

La plateforme IDRA [50] apporte une solution différente aux protocoles habituels en proposant une architecture basée sur des services réseau plutôt que sur des protocoles. Chaque tâche propre à un protocole est assimilée à un service. Il est alors possible de combiner ces services pour reproduire le comportement de chaque pile protocolaire, permettant ainsi à un réseau IDRA de communiquer autant par Bluetooth que par Wi-Fi que par n’importe quel autre protocole.

Dans Uribe-Perez et Pous [149], les auteurs proposent d’aborder l’aspect architectural avec une approche inspirée par la structure du système nerveux du corps humain. Chez l’humain, tous les stimulus sont traités par le cerveau. Toutefois, lorsqu’un neurone sensoriel détecte un danger, le signal peut aussi être envoyé à la moelle épinière pour un traitement plus rapide, ce qui fait que le signal est traité avant même que le cerveau ne le reçoive. De même, les fonctions de base comme respirer et faire battre le cœur n’ont pas besoin du cerveau, étant plutôt la responsabilité de « *medulla oblongata* », une extension de la moelle épinière. Les auteurs proposent d’imiter ce fonctionnement par l’introduction d’intermédiaires intelligents (*smart gateways*) dans le réseau.

Chapitre 2 – État de l’art

Ces intermédiaires conservent leur rôle habituel de collecte des données sensorielles et de transmission vers Internet. Ils acquièrent cependant un second rôle de traitement et de prise de certaines décisions. Cette modification vise à réduire le trafic sur le réseau en éliminant certaines données devenues inutiles, ce qui simplifie ensuite le travail du « cerveau » central de la ville. L’idée est donc que les intermédiaires intelligents agissent au même titre que la moelle épinière. Cette approche ressemble beaucoup à l’informatique de périphérie (*edge computing*) traitée à la section 2.1.2. Ce concept d’intermédiaires intelligents est un sujet actif de la recherche sur les réseaux de capteurs [45] et est un fondement important de cette thèse.

D’autres architectures sont proposées dans Jin et al. [84] et sont comparées et analysées du point de vue de la qualité du service, c’est-à-dire à la confiance qu’un message sera bel et bien reçu par le bon objet aussitôt que possible. Une première architecture est celle où les objets sont connectés entre eux dans un réseau distinct du réseau Internet. L’usager doit donc se connecter directement au réseau d’objets. Un exemple d’usage donné est pour la gestion automatique d’un stationnement. Une seconde architecture consiste à relier le réseau précédant à l’Internet et à faire en sorte que l’usager accède au réseau d’objet via le réseau Internet. Cette seconde architecture est identifiée comme propice à la gestion du trafic routier.

De nombreux travaux portent sur des applications concrètes de l’IdO pour les villes intelligentes. L’une de ces applications est l’établissement d’un protocole efficace pour le contrôle d’une intersection routière pour améliorer le transport dans une ville [115]. D’autres auteurs proposent de virtualiser les intersections en élisant une voiture en attente comme maître d’orchestre chargé d’ordonner le trafic pour une courte période en tirant profit de la communication entre véhicules (VANET) [59, 58].

À la fin de ce chapitre, il sera question des différentes technologies de positionnement. La recherche sur les villes intelligentes inclut aussi un aspect de positionnement pour établir les grandes tendances en mobilité. Ainsi, dans Fernandez-Ares et al. [57], les auteurs utilisent un Raspberry Pi pour détecter les appareils mobiles à proximité en écoutant la bande 2,4GHz pour les signaux Wi-Fi et Bluetooth. Ils déploient leur système dans une discothèque, un bâtiment public et sur une autoroute, prouvant qu’un même système peut être utilisé à différentes échelles. Une caractérisation des déplacements comme celle qu’ils proposent est une étape importante vers l’optimisation de la gestion de déplacement au sein d’une ville intelligente.

Les réseaux intelligents

L’Internet des objets peut être déployé et utilisé à toutes les échelles. Les échelles les plus visibles sont celles des maisons et des villes intelligentes. Mais il existe aussi une échelle globale où ces technologies peuvent être utiles. Les réseaux intelligents (*smart grid*) en sont un exemple. Les auteurs de Fadel et al. [62] présentent une revue des différents usages de l’IdO dans ce secteur.

L’industrie 4.0

Dans les dernières années, il est devenu courant de parler de la quatrième révolution industrielle, ou de l’industrie 4.0. À titre de rappel, les trois précédentes révolutions sont : 1) la mécanisation et la machine à vapeur, 2) la production en chaîne et l’électrification et 3) l’automatisation et l’informatisation [102]. La définition de la quatrième révolution est moins claire, potentiellement parce que la révolution est en cours et sa portée est encore mal définie.

Elle soutient néanmoins que l’industrie 4.0 s’opère dans une fusion des mondes virtuels et réels. Une partie importante de cette fusion est l’utilisation des techniques d’intelligence artificielle et de traitement des mégadonnées pour l’optimisation des processus industriels. Un second aspect est la personnalisation des produits pour répondre au design du client plutôt que l’inverse [98]. Dans certains cas, on parle aussi de réalité augmentée pour prévisualiser des tâches, notamment dans le domaine de la construction navale [64].

À ce titre, l’IdO est à la base même de cette révolution puisque sans cela, il ne serait pas envisageable de collecter à la main toutes les données nécessaires à sa mise en place. Toutes les caractéristiques déjà mises de l’avant par les autres applications sont réutilisées dans le contexte industriel. La technologie de RFID dont il sera amplement question dans la seconde partie de la thèse tire aussi son origine du monde industriel et fait partie des technologies facilitant la transition de la 3^e à la 4^e révolution [49]. Ainsi, l’industrie 4.0 est un domaine d’application possible très intéressant pour le BMP, car, dans une usine, de nombreuses décisions peuvent être prises à partir d’information disponible localement. De plus, de nombreuses usines sont déjà équipées pour faire le suivi d’objet, RFID ayant été, à la base, développée pour cela.

2.1.2 La pérennité de l’IdO

Cette section s’attarde sur deux concepts importants pour l’Internet des objets : l’informatique de périphérie et l’informatique autonome. Ces deux éléments sont considérés comme essentiels pour assurer un bon avenir à l’IdO et servent de base philosophique au BMP.

Informatique de périphérie

L’informatique de périphérie (*edge computing*) est une nouvelle tendance de l’infonuagique qui consiste à traiter ou à prétraiter des données le plus près possible de leur point d’émission. Les buts sont de réduire le volume de données sur le réseau et donc le volume de donnée à traiter sur le serveur central. Les concepts d’informatique de périphérie se rapprochent des concepts de l’informatique de maillage (*mesh computing*), du *grid computing*, du poste-à-poste et de l’informatique autonome. L’un des principaux avantages est de pouvoir sécuriser les données plus près de leur source, ce qui augmente la sécurité du système. Il s’agit avant tout d’une architecture de système répartie cherchant à tirer profit de la puissance de calcul disponible sur chaque intermédiaire entre le capteur et le ou les serveurs centraux [139]. Il est aussi possible d’utiliser les capacités de stockage d’informations de ces ordinateurs périphériques [136].

Informatique autonome

L’informatique de périphérie présentée dans la section précédente forme l’une des bases des systèmes modernes basés sur l’infonuagique. Il y a toutefois un autre concept important à considérer pour ces systèmes : l’informatique autonome. Introduit pour la première fois en 2003 par Kephart et Chess sous le nom « *autonomic computing* » [87], ce paradigme décrit comment les systèmes peuvent se gérer eux-mêmes, en ayant recours à un minimum d’actions humaines. Dans l’informatique autonome, le rôle de l’opérateur humain consiste simplement à donner un objectif haut niveau pour le système et à faire l’entretien des composants physiques. Tous les aspects informatiques sont gérés par le système, de l’intégration de nouveaux composants à leurs optimisations. Tout est conçu pour minimiser les besoins humains.

L’informatique autonome est la plus souvent décrite autour de quatre thèmes centraux, les quatre « autos » (de l’anglais *self*). Il s’agit de l’autoconfiguration, de l’auto-optimisation, de l’auto-réparation et de l’auto-protection. L’autoconfiguration référence principalement l’ajout de composants au système. C’est elle qui décrit comment ces nouveaux composants seront utilisés afin de satisfaire au mieux l’objectif général et la manière dont leur intégration peut se faire de façon transparente pour le reste du système, sans impacter négativement les autres composants. Généralement, il est admis que les nouveaux composants vont contacter un registre afin de s’enregistrer et d’indiquer leurs capacités afin que ceux déjà présents puissent s’accorder. Naturellement, cela implique une connaissance minimale du système autonome pour savoir quel protocole de communication utiliser et sur quel port communiquer.

L’auto-optimisation poursuit un peu dans le même sens que l’autoconfiguration dans l’optique où elle est aussi visée d’effectuer des modifications pour améliorer les performances. L’auto-optimisation est à l’autoconfiguration ce que la formation continue est à la formation scolaire obligatoire. Le système peut, par exemple, effectuer des mises à jour de ses logiciels, ou modifier les tables de routage en fonction de l’évolution du trafic. Toutes les étapes et tous les éléments du système peuvent être modifiés automatiquement par les mécanismes d’auto-optimisation. Un autre aspect important de l’auto-optimisation touche les économies en ressource, plus particulièrement l’efficacité énergétique. C’est d’ailleurs une composante importante que les systèmes d’IdO cherchent à optimiser [138, 137].

Les deux dernières propriétés concernent surtout l’intégrité du système. Ainsi, pour être réellement autonome, le système doit être en mesure de détecter et de corriger par lui-même les erreurs, qu’elles soient logicielles ou matérielles. Le système peut, par exemple, détecter qu’un composant ne répond plus et confier ses tâches à un autre. C’est un élément particulièrement important pour les systèmes utilisant l’Internet des objets puisque les objets sont souvent sujets au manque d’énergie par épuisement des piles.

Il importe alors de pouvoir détecter ces pannes pour rediriger le trafic du réseau et différencier l’absence de données d’une donnée nulle ou neutre afin d’effectuer les bons calculs. Une approche développée au DOMUS consiste à utiliser des réseaux bayésiens pour vérifier les données des différents capteurs d’un environnement intelligent et pour trouver le point d’erreur, le cas échéant [117]. Cette approche est aussi utilisée dans les réseaux sans fil [24] et dans l’infonuagique [55]. Il est aussi possible d’utiliser des réseaux de neurones récurrents pour cette tâche [85]. L’autoprotection concerne principalement les aspects de sécurité du système. Son rôle est de prévenir et de détecter les intrusions malicieuses et d’effectuer les actions nécessaires pour prévenir les interruptions en cascade pouvant provoquer la panne complète du système. Il s’agit donc de gérer les accès, de chiffrer les communications, de configurer les pare-feux, de détecter les vols de données, de rediriger les communications pour éviter les points de surchauffe, etc. Il est ainsi possible de sécuriser un réseau en modélisant les attaques connues pour arriver à les détecter durant leur mise en place et déterminer comment s’en prémunir [32].

Le protocole décrit dans ce document vise à utiliser certains de ces concepts dans son fonctionnement. L’approche ne poursuit pas réellement l’atteinte des quatre autos, mais suit plutôt une trajectoire similaire, tel qu’il sera expliqué dans les prochains chapitres.

2.2 Sur les protocoles de communication

Il existe un protocole de communication pour répondre à chaque besoin, ou presque. Un protocole de communication, c’est une entente sur une façon de communiquer pour deux ou plusieurs objets. Cette entente est établie pour répondre à un besoin spécifique et va être fortement influencée par son environnement. Une liste non exhaustive des caractéristiques d’un environnement contient :

- Est-ce filaire ou sans fil ?
- Est-ce que le médium est propice aux collisions ?
- Est-ce que le médium est prompt à saturer ?

Pour faciliter le développement de protocoles pouvant répondre individuellement à chacune de ces questions et ainsi faciliter la réutilisation des protocoles, la communication informatique se fait habituellement à travers plusieurs couches. Un modèle de couche protocolaire souvent utilisé est le modèle OSI [166], développé pour l’*International Organization for Standardization*. Le modèle comporte sept couches, chacune s’appliquant par-dessus celle qui la précède. Dans le cadre de cette thèse, seules les premières couches sont importantes. La première couche est celle dite « physique ». Elle concerne l’interface avec la technologie physique de communication, soit le type d’antenne radio. La seconde couche est la couche liaison. Elle rassemble principalement les éléments d’accès au médium et d’évitement des collisions. À ce niveau, il est possible de communiquer avec les autres objets connectés sur le même médium. Un protocole bien connu de cette couche est le protocole MAC. La couche réseau est la troisième de la pile. Cette couche inclut les éléments permettant la communication entre les différents réseaux. C’est à ce niveau qu’existe le très connu protocole IP. La couche transport forme la quatrième couche. C’est à ce niveau que les protocoles se préoccupent de la qualité de service et de la segmentation des messages volumineux. C’est la dernière couche pertinente pour la présentation du BMP. Les autres couches concernent principalement les niveaux applicatifs et l’utilisation des messages.

Dans cette section de l’état de l’art, certains protocoles seront présentés. Ils ont été choisis en fonction de leur pertinence historique, de leur usage en IdO et des caractéristiques communes au BMP. Ils ne sont pas groupés par couche, mais plutôt par domaine d’utilisation. Il y a d’abord les protocoles généraux utilisés partout, puis les protocoles sans fil et finalement les protocoles de haut niveau.

Une dernière catégorie possible aurait été les protocoles *ad hoc*. Il a été choisi de ne pas les présenter puisqu’ils présentent tous un besoin de connaître avec quoi chaque objet communique, ce qui, au meilleur de nos connaissances et comme nous le verrons au chapitre 3, n’est pas possible dans le BMP.

2.2.1 Les protocoles généraux

Les protocoles de cette partie sont les plus connus et les plus utilisés. Ils font habituellement partie du cursus obligatoire de tout cours de télématique. Ils seront donc présentés rapidement, en insistant sur les éléments réutilisés par le BMP.

Internet Protocol

Le premier protocole à mentionner est sans nul doute l’Internet Protocol [125], IP, sur lequel le HTTP et le web sont fondés. Dans son expression la plus simple, le protocole n’a que deux fonctions : transmettre un datagramme à une adresse sur un réseau local (adressage) et séparer un trop long message en plusieurs morceaux pour l’expédition (fragmentation). L’IP ne s’occupe toutefois que de gérer les adresses, sans se soucier du nom que peut posséder l’adresse (DNS) ni du chemin que le message doit prendre pour se rendre à destination (ARP). Dans la version 4 (IPv4) [125], les adresses se composent de quatre octets, où le premier indique le réseau et les trois derniers l’adresse locale. L’adressage est absolu, dans le sens où il ne peut y avoir deux endroits ayant la même adresse. L’en-tête d’IP inclut quelques éléments intéressants, comme un numéro de version, une somme de contrôle et de l’espace pour quelques drapeaux.

Chapitre 2 – État de l’art

Un autre élément important est le *Time to Live*. Ce champ indique le temps maximal que le message peut rester dans le réseau, exprimé en secondes. Toutefois, puisque chaque module par où le message transite doit réduire ce temps d’au moins 1, il s’agit plutôt d’une borne maximale de temps qu’un temps absolu. En effet, les réseaux sont typiquement assez performants pour que le traitement dure moins d’une seconde.

Considérant l’augmentation exponentielle du nombre d’appareils connectés au réseau IP, l’adressage sur quatre octets est devenu insuffisant pour identifier de façon unique tous les appareils. Une autre version du protocole, nommée IPv6, a donc vu le jour. Parmi les changements apportés, on note que la taille de l’adresse est passée de 4 octets à 16 octets. Ainsi, il est virtuellement impossible de manquer d’adresses, considérant la taille des appareils et la taille de la Terre. Dans leur livre, Kurose et Ross vont même jusqu’à dire que grâce à IPv6, chaque grain de sable de la Terre peut avoir sa propre adresse IP [96].

IP permet aussi d’envoyer des messages en mode diffusion (*broadcast*), c’est-à-dire à l’ensemble du réseau, sans adresse de destination précise. Pour ce faire, l’adresse de destination 255 255 255 255 est utilisée (pour IPv4).

Il existe de nombreux algorithmes de diffusion. Un premier, nommé *N-way-unicast*, consiste à ce que l’objet initial envoie un message directement à chaque autre objet du réseau dont il connaît l’adresse. Le danger est alors que la duplication ne s’arrête jamais et paralyse le réseau. De nombreuses solutions existent.

Un algorithme simple consiste à n’accepter un message diffusé que de son plus proche voisin *unicast*, comme établi pour les algorithmes de routage. Cette méthode est appelée *reverse-path-forwarding*. La méthode *sequence-number-controlled flooding* requiert pour sa part que chaque nœud conserve en mémoire une liste des messages diffusés déjà transmis de façon à pouvoir les détecter et éviter de les retransmettre.

L’Internet Protocol est utilisé par de nombreuses applications. C’est d’ailleurs sur lui que reposent les protocoles UDP et TCP, les deux plus connus.

User datagram Protocol

Le *User Datagram Protocol* [121], UDP, est l’un des protocoles les plus simples faisant usage de l’IP pour sa couche réseau. Tout ce qu’il offre concrètement de plus que l’IP est l’adressage sur des ports précis. Une image représentant bien la différence est celle d’une boîte postale. L’IP renseigne sur l’adresse à laquelle une lettre doit se rendre, alors qu’UDP renseigne sur la personne à l’adresse à qui la lettre est adressée.

Transmission Control Protocol

UDP est souvent référencé en tant que protocole sans connexion, par opposition au *Transmission Control Protocol* [126], TCP, qui lui requiert une connexion. Cette connexion prend la forme d’un échange de salutations (*handshake*) entre la source et la destination. Il est à noter que cet échange se fait seulement entre les objets terminaux, les différents intermédiaires (routeurs, commutateurs ...) n’y prenant pas part. Cette connexion fait de TCP un protocole point à point où les échanges peuvent se produire dans les deux sens, en *full-duplex*. Il n’est pas possible avec TCP de faire des communications *broadcast* ou *multicast*. Comme son nom l’indique, TCP offre des mécanismes de gestion de flux, dans le but de réduire les collisions entre datagrammes. Il garantit aussi la livraison des messages, dans l’ordre et dans leur intégralité, sans erreurs de transmission possibles. Pour ces raisons, la majorité des données sensibles circulent via TCP. Le protocole présenté dans ce document ayant peu à voir avec TCP, la description des mécanismes permettant à TCP d’offrir ces garanties ne sera pas faite plus en détail.

2.2.2 Les sans-fils

Cette section regroupe certains des protocoles les plus en vogue dans le domaine de l’IdO.

Wi-Fi

Le Wi-Fi est un terme générique représentant une famille de protocoles sans fil des couches liaisons et physiques regroupées sous la norme IEEE 802.11. Au fil des années, de nouvelles versions se sont ajoutées afin d’augmenter la vitesse de transmission, la portée et le nombre de connexions simultanées. Ces améliorations se font grâce à de nombreuses modifications, principalement matérielles. Ainsi, il est possible de choisir une fréquence d’émission plus ou moins haute pour moduler la force et la portée du signal. Il est aussi possible d’ajouter des antennes afin d’augmenter le nombre de canaux concurrent et ainsi augmenter la vitesse de la communication.

Le plus souvent, le Wi-Fi sert à créer un réseau sans fil local, un WLAN (pour *Wireless Local Area Network*). Typiquement, un point d’accès Wi-Fi sera positionné de façon à optimiser la portée du signal et sera branché via une connexion filaire à un routeur donnant accès à Internet.

C’est à ce point d’accès que se connectent les appareils souhaitant accéder à Internet. Il est alors nécessaire que chaque appareil établisse une connexion avec le point d’accès. Cette étape est plus complexe à réaliser lorsqu’un même WLAN fournit plus d’un point d’accès pouvant se chevaucher. Puisque le Wi-Fi requiert que chaque point d’accès publie régulièrement son adresse MAC et son SSID, la solution consiste généralement à attendre un peu avant d’établir une connexion puis de choisir le point d’accès offrant le signal le plus puissant.

Il n’est pas rare que plusieurs appareils soient connectés à un même point d’accès en même temps. Tout comme pour les réseaux filaires, cette situation peut entraîner de nombreuses pertes de messages si les appareils communiquent tous en même temps. Pour cette raison, Wi-Fi propose sa propre couche MAC, CSMA/CA (pour *carrier sense multiple access with collision avoidance*). Tout comme pour Ethernet le fait avec les connexions filaires, CSMA utilise le hasard pour tenter de réduire les collisions. Simplement, le protocole attend que le canal de transmission soit libre puis envoie un court message qui devra être confirmé avant que le vrai message soit envoyé et confirmé. Si un message n’est pas confirmé, il faut attendre un temps aléatoire avant de réessayer [96, p. 533]. Un autre mode consiste à demander l’autorisation d’émettre. Dans ce mode, un appareil va envoyer un très court message (Request to Send, RTS) au point d’accès pour demander la permission d’envoyer un plus gros message en indiquant le temps requis. Le point d’accès répond alors à tous avec une autorisation (Clear to Send, CTS). Le demandeur sait alors qu’il peut envoyer pendant que les autres reçoivent le CTS comme une consigne d’attendre la confirmation de la fin du transfert. RTS et CTS étant de courts messages, les risques de collision sont plus faibles.

Le Wi-Fi peut aussi être utilisé pour créer des réseaux ad hoc où il n’y a pas de points d’accès. Ce type de réseau est utile pour communiquer entre appareils situés à proximité ou pour les endroits ne disposant pas d’infrastructure particulière.

Deux versions de Wi-Fi existent aussi spécifiquement pour l’IdO. Il s’agit des versions 802.11AF et 802,11 AH. Elles opèrent à des fréquences plus basses que le Wi-Fi domestique, à 900MHz au lieu de 2,4GHz. 802.11AH, nommée Halow, consomme très peu d’énergie, ce qui lui permet de concurrencer des protocoles comme LoRa et Zigbee [129].

Les protocoles domotiques

La section précédente présente sommairement la famille de protocoles IEEE 802.11, le Wi-Fi. Cette famille vise à créer un protocole sans fil capable de remplacer les réseaux filaires de haute capacité. Les communications à faible débit et à faibles portées sont quant à elles laissées à la famille IEEE 802.15. Elle définit plusieurs protocoles connus, tels Bluetooth, Zigbee et UWB. Tous ces protocoles ont pour but de permettre la création de réseaux sans fil personnels, WPAN (pour *Wireless Personal Area Network*).

Bluetooth a d’abord été créé par la compagnie Ericsson en 1994. Par la suite, il a été intégré à IEEE 802.15.1. Le standard est maintenant maintenu par le *Bluetooth Special Interest Group*. Le protocole fonctionne selon un système maître/esclave, où un appareil est choisi pour agir en tant que maître et peut regrouper jusqu’à sept esclaves dans un réseau restreint nommé Piconet. À tout moment, les appareils peuvent s’entendre pour changer de maître. Dans ce modèle, les esclaves ne peuvent pas communiquer entre eux, mais seulement avec le maître.

Dans ce protocole, les messages sont encodés avec une clé AES de 128 bits pour offrir une sécurité minimale aux utilisateurs. Bluetooth 5.2 est la dernière version disponible et offre une vitesse de transfert de 50Mbit/s sur une distance maximale de 240m⁶.

Bluetooth Low Energy (BLE) vise à offrir un WPAN de portée semblable à Bluetooth en utilisant moins d’énergie. La technologie vise donc à être adoptée par l’Internet des objets, où l’efficacité énergétique revêt une grande importance afin d’augmenter la durée de vie des objets utilisant une pile. La portée de BLE est de plus de 100 mètres, pour une vitesse de transfert allant jusqu’à 2Mbit/s. Il est possible d’utiliser Bluetooth en mode diffusion et maillage. Depuis la version 5.1, Bluetooth intègre une mesure de l’angle d’arrivée et de départ des messages sur l’antenne, facilitant le suivi de la position des objets [145].

⁶ <https://en.wikipedia.org/wiki/Bluetooth>, consultée le 22 juillet 2020

La norme IEEE 802.15.3 existe afin de fournir des WPAN à haute fréquence. UWB [118] est l’un de ses représentants les plus connus. Le protocole vise à offrir une communication à faible énergie de courte portée sur une large bande de fréquence du spectre électromagnétique. Cette large bande permet, entre autres, de réduire le problème des chemins de propagation multiple (*multipath propagation*) puisqu’il devient alors probable qu’au moins l’une des bandes puisse trouver un chemin direct entre l’émetteur et le récepteur.

De son côté, la norme IEEE 802.15.4 [70, 167] vise à encadrer les WPAN à basse fréquence. Cette norme spécifie les niveaux physique et liaison qui servent de fondations à d’autres protocoles de plus haut niveau, tel Zigbee. Zigbee a été conçu pour offrir des services de communication à faible portée, à faible fréquence et à faible consommation énergétique. Les communications sont chiffrées via un chiffrement à clé symétrique de 128 bits. La portée varie de 10 mètres à 100 mètres en ligne droite, bien qu’il soit possible d’augmenter la portée effective en utilisant un maillage d’objets Zigbee. Un réseau Zigbee contient trois types de nœuds. Il y a tout d’abord un et un seul coordonnateur puisque c’est celui qui initie le réseau. Ce nœud, généralement le plus puissant, est le seul pouvant établir des liens avec d’autres réseaux. Il est responsable de sauvegarder les informations concernant le réseau et agit en tant que nœud de confiance pour l’entreposage et l’attribution des clés de chiffrement. Ensuite, il y a les nœuds de routage. En plus d’héberger une application Zigbee, ils vont aussi servir de relais entre les autres nœuds, pour créer le maillage. Finalement, il y a les nœuds de fin (*Zigbee End Device*). Ces nœuds peuvent communiquer avec leur parent (le coordonnateur ou un routeur) uniquement. Ce sont généralement les capteurs de plus bas niveaux. La communication fonctionne principalement sur le mode *unicast* lorsque l’adresse est connue, sinon par diffusion pour déterminer l’adresse. Zigbee est limité à 65 000 appareils dans un même réseau [143].

Dans la même catégorie de protocoles que Bluetooth et Zigbee, on retrouve le protocole propriétaire Z-Wave [61]. Z-Wave est aussi conçu pour faciliter la création de réseaux d’objets sous la forme d’un maillage. Un aspect intéressant de Z-Wave est que tous les appareils Z-Wave sont conçus pour être interopérable et pour être utilisé dans le même réseau, peu importe le fabricant de l’objet. Un réseau Z-Wave contient au maximum 232 objets, mais il est toutefois possible de relier entre eux plusieurs réseaux pour contourner cette limitation. La portée est d’environ 30 mètres, pour un débit variant entre 9,6 et 100 kbit par seconde. Tout comme avec Zigbee, les communications sont chiffrées avec une clé symétrique AES de 128 bits.

Insteon [103] est un autre protocole privé visant l’automatisation de la maison. Dans Insteon, chaque objet est un pair capable de recevoir, émettre et transmettre des messages. Insteon est toutefois différent puisqu’il permet de communiquer à la fois en mode sans fil et en mode filaire, via le câblage électrique de la maison, similairement à X-10 [31]. Sa capacité est toutefois très limitée, à 14 octets par message. Un message ne peut aussi être retransmis que trois fois.

Similairement, KNX est un standard européen pour la maison autonome pouvant communiquer à la fois par le câblage électrique de la maison et par les ondes électromagnétiques [151]. KNX permet d’organiser les objets avec une topologie en arbre, en ligne ou en étoile. KNX supporte jusqu’à 57 375 objets en utilisant un adressage de 16 bits⁷.

Utilisant la norme 802.15.4, concurrençant Zigbee, on retrouve 6LoWPAN, qui vise à faire de la communication à basse fréquence et à faible énergie en utilisant IPv6. Sa topologie est aussi sous la forme d’un maillage [112]. Construit par-dessus 6LoWPAN, Thread est un protocole visant à garantir un très haut niveau de sécurité. Thread reconnaît huit niveaux d’objets dans son réseau, selon leur capacité à communiquer avec l’extérieur et leur puissance de calcul [150].

⁷ <https://www.knx.org/knx-en/for-professionals/What-is-KNX/A-brief-introduction/index.php>, consultée le 22 juillet 2020

Longue portée

Les protocoles présentés dans la section précédente visent tous une portée effective de l’ordre d’une maison, soit des PAN (*personal area network*) ou des LAN (*local area network*). Cette section va introduire certains protocoles ayant une portée plus grande, de l’ordre du kilomètre. Ce sont des WAN (*wide area network*), visant un réseau à l’échelle d’une ville, voire d’un pays.

Traditionnellement, les réseaux de téléphonie sans fil ont été utilisés pour transmettre des données textuelles sous forme de SMS par les objets urbains. Puis, avec les réseaux 3G et LTE, le contenu s’est complexifié pour inclure tout ce qui est permis par les MMS. Le futur réseau 5G promet de révolutionner le domaine. Ces réseaux n’ont toutefois pas été considérés pour cette thèse.

Plus récemment, des réseaux sans fil longues portées ont été spécialement conçus pour l’IdO. Les plus populaires sont le Lora et le Sigfox. Ces réseaux n’étant pas déployés au Québec, ils n’ont pas plus été considérés pour la couche physique du BMP. Toutefois, Lora présente une caractéristique intéressante dans sa façon de limiter le trafic sur le réseau. À l’origine, un objet communiquant via LoRa était limité à transmettre moins de 0,1 % à 1 % du temps, selon la fréquence radio. Une limitation de 1 % signifie donc que si un message prend 1 seconde à envoyer, l’objet ne peut plus rien transmettre pendant les 99 prochaines secondes. Cette limitation a pour objectif de permettre à un grand nombre d’objets de communiquer avec une même antenne relais. Puisque LoRa est prévue pour une très grande portée, il est plus que probable qu’un grand nombre d’objets seront dans la portée de chaque antenne [8].

Protocoles réflexes

Cette dernière section sur les protocoles sans fil s’intéresse à deux protocoles qui ne permettent pas une réelle communication entre objets, mais plutôt d’accéder à des données « statiques » de manière sans fil. Deux protocoles sont d’intérêts pour cet usage : Near Field Communication (NFC) et Radio Frequency Identification (RFID). NFC est très directement inspiré de RFID. Il permet d’enregistrer un court texte sur une puce qui pourra ensuite être lue par un lecteur NFC. La portée de la lecture est généralement inférieure à 10 cm, ce qui en fait une technologie idéale pour les cartes d’accès. Le protocole a gagné en popularité lorsqu’il a été intégré dans les téléphones intelligents puis dans les cartes de paiement sans contact (*paypass*) [42].

La RFID [38] fonctionne de façon similaire, mais à une portée pouvant atteindre plusieurs mètres. Une puce RFID ne contient habituellement qu’un numéro d’identification qu’elle émet sur demande. Il existe deux types de puces : les puces actives et les puces passives. Les puces actives possèdent leur propre source d’énergie, ce qui leur permet d’émettre d’elles-mêmes. Les puces passives vont plutôt utiliser l’énergie contenue dans une requête envoyée par une antenne RFID pour s’activer et répondre avec leur identifiant. Leur portée est donc moins grande que celle des puces actives. RFID est la technologie de positionnement utilisée dans cette thèse et sera présentée plus en détail dans les articles de la seconde partie de la thèse.

2.3 Sur les technologies de positionnement

Puisque le protocole présenté dans cette thèse fait de la position des objets son thème principal, cette partie de l’état de l’art va présenter les grandes méthodes de positionnement d’objets. Il existe trois principales façons sans fil de localiser un objet : le temps de vol d’un signal, l’angle de réception d’un signal et la force de réception d’un signal. Le signal consiste généralement en une onde électromagnétique, bien que dans certains cas un signal sonore (ou ultrasonore) soit utilisé [158]. La littérature distingue habituellement deux scénarios pour le positionnement, selon si l’objet à localiser se trouve à l’extérieur ou à l’intérieur.

2.3.1 Positionnement extérieur

Le positionnement extérieur est celui avec lequel tout le monde est plus ou moins familier. Le système GPS (et ses concurrents) fait partie intégrale de nos vies, étant présent dans tous les téléphones intelligents et dans la majorité des voitures récentes. Il fonctionne en utilisant le temps de vol d’un signal électromagnétique entre un satellite en orbite et l’objet récepteur. Le temps de vol est converti en distance. Lorsque trois ou plus mesures de distances d’autant de satellites sont obtenues, l’objet détermine alors ses coordonnées spatiales (latitude, longitude et altitude) en croisant ces distances avec les positions des satellites tirées des éphémérides. La précision de base du système est de trois à cinq mètres. Elle peut être augmentée en couplant les signaux satellitaires avec ceux d’une station terrestre, pour atteindre une précision millimétrique. Le GPS demande toutefois que l’objet soit en ligne de vue directe avec les satellites, ce qui rend son utilisation complexe, voire impossible, pour les scénarios intérieurs ou dans les zones urbaines de forte densité [122]. Le GPS a la particularité d’être la seule méthode de positionnement à être universelle et gratuite pour tous. Lorsque le ciel est visible, la technologie n’est disponible avec rien de plus qu’un simple récepteur pour équipement.

2.3.2 Positionnement intérieur

Le positionnement intérieur est plus complexe à réaliser. Dans la majorité des cas d’utilisation, l’objet à localiser n’est pas en ligne de vue de l’antenne devant émettre le signal. Aussi, les antennes ne sont souvent pas assez puissantes pour couvrir l’ensemble de l’environnement avec leur signal, ce qui peut créer des zones inaccessibles. De plus, les signaux sont souvent réfléchis ou réfractés par les surfaces rencontrées, ce qui peut influencer la force des signaux. De même, il est possible qu’un même signal atteigne plusieurs fois un même objet, causant des problèmes d’isolation de la mesure du temps, de la puissance ou de l’angle.

Les technologies les plus courantes de positionnement intérieur sont le Wi-Fi, la RFID, l’UWB et le Bluetooth. Dans tous les cas, les algorithmes de positionnement sont les mêmes, adaptés pour les caractéristiques propres à la technologie. On distingue deux catégories de méthodes : celles utilisant une mesure du signal pour calculer une position (on parle alors de trilatération ou de triangulation) et celles comparant la mesure à des mesures préenregistrées (en anglais, on parle alors de *fingerprinting*). Les contributions sur ce sujet présentées dans la seconde partie de la thèse entrent dans la seconde catégorie. Une description détaillée des différentes méthodes présentes dans la littérature est donnée dans l’introduction des articles de ces chapitres.

Depuis quelques années, des chercheurs cherchent à intégrer directement le positionnement dans les protocoles de communication. Ainsi, un article de Seol et al. [141] explore à la fois l’efficacité de la couche MAC et RFID et une méthode de positionnement pour des puces passives. Ils comparent l’efficacité de deux algorithmes de la couche MAC, Aloha avec créneaux (*Framed Slotted Aloha*) et un parcours d’arbre (*Tree-walking polling*), à détecter correctement un grand nombre de puces RFID en même temps. Ils concluent que la méthode Aloha est plus efficace, particulièrement parce qu’il est plus rapide à détecter les puces, ce qui est important pour le suivi de position qu’ils cherchent à obtenir via leur méthode de positionnement.

Cette méthode est plutôt simple, puisqu’elle consiste simplement à compiler les antennes RFID détectant une puce à chaque moment pour ensuite déterminer dans quelle zone se trouve la puce, connaissant le rayon de détection maximal des antennes. Il s’agit donc d’une méthode dite *Range-Free*, puisqu’aucun calcul de distance n’est nécessaire. La méthode n’utilise donc qu’une notion de distance maximale à partir d’un point connu pour positionner les objets.

DEUXIÈME PARTIE

CONTRIBUTIONS EN COMMUNICATION

« It doesn't have to be like this
All we need to do is make sure we keep talking »

— « Keep Talking », *The Division Bell*, Pink Floyd, 1994

Lors de la présentation de l'état de l'art, de nombreux protocoles de communication ont été introduits. Parmi ceux qui sont orientés pour l'IdO, Bluetooth et Z-Wave sont les ensembles de protocoles les plus populaires. Les versions les plus récentes ont apporté un grand soin à augmenter l'autonomie énergétique des appareils tout en augmentant le débit disponible. De plus, il est maintenant possible de créer des constellations d'objets dans la plupart des protocoles. Un désavantage demeure toutefois : la nécessité d'établir un réseau et de connaître l'identité d'au moins un autre objet du réseau pour pouvoir communiquer. Dans un réseau d'objets immobiles, ce n'est pas réellement un problème. Lorsqu'on y ajoute des objets mobiles toutefois, il peut alors devenir compliqué de se connecter au réseau. Si le but du réseau est de communiquer des informations critiques ou privilégiées, ce coût de connexion vaut sans doute la peine et les protocoles actuels sont tous indiqués, d'autant plus s'ils offrent le chiffrement bout à bout des données.

Pour transmettre de simples données factuelles, ces étapes peuvent toutefois sembler superflues. En particulier, cette thèse avance que les étapes de création et de connexion à un réseau ne sont pas toujours utiles. Il existe de nombreux cas, dont certains seront présentés dans les prochains chapitres, où il serait suffisant de simplement transmettre l'information à quiconque peut la recevoir. L'idée derrière le protocole BMP est un peu celle de la distribution de feuillets publicitaires dans une foule : une personne se tient à un point stratégique et distribue son information à tous ceux qui passent à proximité, sans pour autant chercher à établir un contact permanent.

Cette analogie est intéressante pour un autre point névralgique du BMP, celui de la décentralisation. Dans une certaine mesure, on peut considérer le distributeur comme étant un agent indépendant. Considérons aussi sa pile de feuillets comme étant sa réserve d'énergie. Alors, le rôle du distributeur est de donner son information à tous tant qu'il lui reste de l'énergie, de façon totalement indépendante. D'autres distributeurs peuvent être situés à d'autres endroits de la même foule et agir de la même manière, pour créer un ensemble distribué.

Aucun n'a besoin d'être connecté aux autres pour faire sa tâche. Ils peuvent toutefois bénéficier à se communiquer de l'information, sur le taux de distribution par exemple, afin de mieux coopérer pour l'atteinte d'un but commun. L'analogie n'est cependant pas complètement appropriée pour décrire le BMP, puisqu'elle nécessite un élément central que le BMP n'a pas, à savoir le point central d'origine des feuillets.

Pour conclure sur la notion de réseau, il importe d'apporter la distinction entre le réseau et le médium. Le médium, c'est le milieu dans lequel une communication a lieu. En général, le médium fait référence au spectre électromagnétique. Les protocoles présentés dans cette thèse utilisent tous ce spectre, soit sous la forme d'onde, soit sous la forme d'un courant électrique. Le réseau est ce qui est construit par-dessus. C'est la résultante d'objets communiquant selon un protocole donné. Dans le cas de cette thèse, un réseau impliquera aussi des notions d'appartenance à un ensemble formant un réseau. Le BMP n'ayant aucune notion d'appartenance, il sera alors dit qu'il n'y a pas de réseau, seulement un médium, par opposition à cette définition. La section 3.1 va s'attarder sur le modèle de communication proposé.

La décentralisation de l'information est un élément important du BMP. Tout le protocole est conçu pour faire en sorte qu'il soit contre nature de concevoir un système centralisé en l'utilisant. Les systèmes centralisés ont historiquement toujours été très populaires, malgré leurs faiblesses. L'une de ces faiblesses est la résilience aux pannes. Le point central de ces systèmes, par définition, n'est pas résilient aux pannes. Il est bien entendu possible de le répliquer et d'introduire de la redondance. Ces étapes sont maintenant accessibles à tous via les gestionnaires de conteneur virtuel comme Docker¹. Il faut toutefois mettre les éléments en place et effectuer de la maintenance, ce qui a un coût. Le système central sert aussi de point de traitement pour les données.

¹ <https://www.docker.com>

Ce point aura habituellement une vue d'ensemble sur l'ensemble du système. Bien que cela permette un traitement très approfondi des données collectées par le réseau d'IdO connecté, ce paradigme peut vite être submergé par les données. En particulier, il faut s'assurer de pouvoir traiter l'ensemble des données plus rapidement qu'elles sont produites pour éviter de prendre un retard qu'il ne sera pas facilement possible de rattraper. De nombreux algorithmes ont été créés pour permettre une bonne agrégation de vastes quantités de données ou pour rapidement en extraire l'information pertinente à une tâche. Ces algorithmes ne sont cependant pas toujours génériques et peuvent mal s'appliquer à certaines situations plus particulières. C'est, entre autres, le cas pour les systèmes où chaque donnée est importante et peut justifier l'envoi d'une commande par le système central vers un élément de la périphérie.

La décentralisation présente certains avantages facilement opposables aux désavantages de la centralisation [131]. Le plus évident concerne la résilience du système qui ne présente plus aucun point unique de panne. Des composants peuvent flancher sans pour autant compromettre l'ensemble. Les données sont aussi fragmentées entre différents secteurs, ce qui réduit leur volume à chaque point. D'un autre côté, construire un système décentralisé demande plus d'effort. Il y a généralement plus de code à écrire. Ce code est souvent redondant si les différents secteurs ont des tâches connexes. Dans ce cas, il y aura aussi une redondance globale dans les données puisqu'une même information pourra être utile à plus d'un endroit au même moment. Cette approche favorise toutefois la présence et l'émergence de comportements locaux. Chaque section du système décentralisé peut avoir ses propres politiques. De plus, ces politiques ne sont pas obligées d'être en accord avec celles d'autres secteurs et peuvent s'adapter directement aux besoins de l'environnement.

L'infonuagique et la virtualisation représentent une nouvelle façon plus hybride de concevoir de vastes systèmes informatiques. Le nuage se présente comme un service capable d'abstraire toutes les notions de redondance et de distribution des calculs. La montée en charge d'un système peut être gérée automatiquement sans qu'un opérateur ait à intervenir. Les calculs peuvent se faire de manière distribuée ou centrale sans que le programmeur doive s'en préoccuper puisque même les adresses de communication peuvent être virtualisées. Toutefois, la puissance quasi infinie de l'infonuagique vient à un terrible prix, du moins pour certaines utilisations. Tout d'abord, ces plateformes sont habituellement la propriété d'une tierce partie. Le système abandonne alors sa souveraineté sur ses données et ses outils de calculs en payant plutôt un droit d'accès qui, à quelques considérations légales près, peut être retiré au bon vouloir de la tierce partie. Ensuite, le fardeau de la sécurité et l'intégrité des données est partagé avec la tierce partie. De plus, la tierce partie n'est pas toujours capable d'indiquer clairement où dans le monde physique les serveurs sont situés. Il y a alors un risque que le gouvernement du pays où sont situées les données puisse y avoir accès pour différentes raisons. Ce gouvernement peut aussi ordonner la suspension du service d'infonuagique. C'est notamment le cas des États-Unis où il est possible de demander l'accès aux données en cas de soupçons de terrorisme. De plus, les États-Unis peuvent interdire tout lien d'affaire avec une compagnie étrangère. Ce sont donc différents éléments à considérer avant d'introduire l'infonuagique dans son système informatique. Au contraire, un système utilisant un BMP ne pourra envoyer ses données ailleurs dans le monde sans violer les principes du protocole. Ainsi, l'information reste locale et les programmes restent sur les objets, ce qui évite d'en partager la propriété et le contrôle.

Les principaux désavantages de l'infonuagique apportent bien la question de la mouvance des données. En effet, certaines données ne devraient-elles pas rester locales ? Est-ce vraiment nécessaire d'envoyer les données vers un autre ordinateur situé arbitrairement loin dans le monde ? Chaque fois que de l'information voyage vers une nouvelle destination, c'est un point d'attaque de plus qui apparaît dans le système. Mais, ces risques sont-ils nécessaires ? Les décisions ne devraient-elles pas être prises dès que toutes les données nécessaires sont disponibles ? Ces décisions ne pourraient-elles pas être prises dans le voisinage géographique immédiat du point de collecte des données. À titre d'exemple, les décisions concernant la température d'une unité d'habitation ne devraient-elles pas être prises au sein même de cette unité ? Les données de température ont-elles alors la moindre raison de sortir de l'environnement immédiat de l'unité ? La section 3.4 présente le mode de gestion des données proposé. Ces questions forment la base de la réflexion dont cette thèse est le produit final.

Le *Bounded Message Protocol*, BMP, est la solution proposée par le candidat pour répondre à ces questions, dans leur version plus formelle présentée dans le chapitre 3. Cette partie de la thèse se découpe en quatre chapitres. Le premier va présenter dans le détail le protocole et ses caractéristiques. Le deuxième chapitre présente les pistes à suivre pour implémenter un BMP. Le troisième chapitre va présenter une étude de la performance du protocole dans des conditions contrôlée en laboratoire. Des objets sont dispersés au sein du DOMUS et la performance est évaluée en matière de messages délivrés selon la densité d'objets et la fréquence des messages. Le dernier chapitre, qui forme la conclusion de cette partie, va préciser les conditions d'utilisations prévues du protocole et introduire certaines applications intéressantes qui pourraient en être faites.



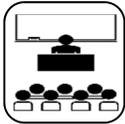
TROISIÈME CHAPITRE

Proposition du BMP



3.1 Modèle de propagation

3.2 Réseau filaire



3.3 Portée des données

3.4 Modulation de la puissance d'émission des capteurs

3.5 Gestion du temps

3.6 Entreposage des données

3.7 Format des messages

3.8 Retour sur les objectifs

Proposition du BMP

L'une des caractéristiques fondamentales de l'humain est sa capacité à créer des outils. Ces outils permettent ensuite la création d'objets toujours plus complexes. Les objets qui nous entourent sont maintenant capables de communiquer entre eux. Pour ce faire, il existe de nombreux protocoles de communication. Toutefois, la plupart demandent la présence d'un réseau et la connaissance des autres objets. Ce modèle n'est pas toujours approprié. Le BMP cherche à créer un autre modèle où la position des objets seulement compte, et pas leur appartenance à un réseau.

Dans l'introduction de cette partie de la thèse, certaines caractéristiques souhaitables pour un nouveau protocole de communication pour l'Internet des objets ont été introduites. L'idée générale est de limiter la propagation de données d'intérêt général à une zone de pertinence décidée par l'émetteur à l'origine de chaque communication. Pour y arriver, chaque message se voit attribuer une distance physique maximale à franchir, de même qu'un temps de vie maximal. Il est donc supposé que chaque point d'émission de messages connaisse sa propre position géophysique et soit en mesure de déterminer lui-même la portée utile des données à transmettre. Un appareil recevant un message doit aussi être en mesure de déterminer si la portée est dépassée ou si la durée de vie est expirée. Le cas échéant, l'appareil doit alors supprimer le message, sans le retransmettre.

Une telle approche vise à réduire la quantité des données présentes dans un habitat intelligent. En effet, en supposant un mode de transmission multi-sauts à faible portée, une section donnée d'un environnement intelligent ne devrait recevoir d'une autre section du même environnement que les données qui la concernent, lui évitant de devoir faire le tri elle-même. De plus, ce mode de communication permettrait de réduire la complexité des algorithmes d'inférence en limitant le nombre de données de bas niveau qu'ils ont à traiter. Plus encore, certains traitements de vérification de la proximité et de la fraîcheur des données habituellement effectués dans les algorithmes d'inférence ne sont plus nécessaires lorsque le BMP est utilisé pour la communication.

Ainsi, le BMP ouvre la porte à une nouvelle classe d’algorithme où les questions de position et de récurrence peuvent être directement intégrées dans l’algorithme. Cette perspective sera explorée plus en détail dans le chapitre 6.

Pour l’ensemble de cette partie de la thèse, il est supposé qu’il n’y a aucun objet malicieux dans l’environnement. Concernant les données échangées, le soin de les chiffrer est laissé à l’application. Concernant l’utilisation du médium et le respect du fonctionnement du protocole, il est supposé que tous les objets agissent comme de parfaits gentlemen, en respectant toutes les règles. Les méthodes de protection face aux objets malicieux ne sont pas explorées dans cette thèse.

Pour la suite de ce chapitre sur les concepts fondamentaux du BMP, l’idée générale sera développée en suivant certains thèmes. Ces thèmes théoriques sont : le mode de diffusion des messages, la portée physique des messages, la portée temporelle et l’entreposage interne aux protocoles des messages. La présentation de l’en-tête du BMP conclut le chapitre en permettant un retour plus pratique sur les différents thèmes.

3.1 Modèle de propagation

Le besoin d'un nouveau protocole a été défini en début de chapitre. Cette section va décrire un aspect important du nouveau protocole BMP. Une façon intuitive de décrire un protocole de communication est de commencer par décrire comment les messages sont envoyés, relayés et reçus par les différents éléments du réseau.

Dans le cas du BMP, le premier point d'une telle démarche est d'indiquer qu'il n'y a pas de mécanismes pour construire, optimiser, entretenir un réseau structuré (comme par exemple, les maillages), seulement un médium de communication. Le BMP fonctionne sur le principe que chaque objet intelligent vit indépendamment des autres, sans avoir aucune conscience de leur existence. Il n'y a donc aucune notion d'introduction de nouveaux éléments ni de routage ni de connexion. Chaque objet ignore totalement s'il est seul dans son environnement ou pas. Le seul moment où un objet sait qu'il n'est pas seul, c'est lorsqu'il reçoit un message. Malgré tout, l'objet continue alors de se croire seul. En effet, du point de vue d'un objet, recevoir un message d'un autre objet à l'instant T n'implique en rien que cet autre objet existe encore au temps $T+1$ ou existait déjà au temps $T-1$. Il serait abusif de faire cette supposition puisqu'à tout moment un objet peut cesser d'exister pour une raison quelconque, l'épuisement de sa réserve d'énergie étant une raison fréquente. Alors que certains protocoles vont envoyer des messages entre les objets à intervalle régulier pour s'assurer de la disponibilité des objets connus, le BMP fait le choix d'ignorer ces besoins et de considérer chaque objet comme étant toujours seul au monde.

Pourtant, le BMP envoie des messages avec l'espoir d'être entendu. Plus encore, ces messages doivent pouvoir être reçus et retransmis. En l'absence de notion de réseau, il n'est pas possible de donner une adresse de destination pour son message. Il n'est donc pas possible non plus de s'adresser à un objet en particulier. Ainsi, toute communication doit être faite à tous les objets présents et à l'écoute sur le médium. Chaque objet recevant un message doit aussi le retransmettre à la ronde en espérant ainsi atteindre d'autres objets qui n'ont pas été atteints par le message d'origine.

De fait, toute communication sans fil fonctionne déjà sur ce principe. En effet, à moins de posséder une antenne directionnelle, il n'est pas possible de contrôler qui peut « lire » les ondes électromagnétiques. La différence entre le BMP et les protocoles à adresses se situe plutôt sur le critère d'arrêt de la retransmission des messages. Avoir un critère d'arrêt de retransmission est tout aussi primordial dans un protocole de communication qu'un critère d'arrêt est nécessaire dans une fonction récursive : sans cela, il ne saurait y avoir de fin. Dans les protocoles à adresses, le critère de fin est typiquement l'atteinte de l'adresse de destination (ou un TTL pour IP). Cela est possible en raison de la présence de tables de routage faisant en sorte que chaque objet connaisse au moins partiellement le chemin pour atteindre chaque autre objet. Il ne retransmet alors un message que s'il est sur le chemin pour la destination, sinon il l'ignore.

N'ayant pas d'adresse, le BMP va plutôt s'appuyer sur les positions relatives des objets. Il y aura donc une distance maximale pouvant être franchie par un message au-delà de laquelle tout objet recevant le message devra le détruire. Ainsi, un objet souhaitant envoyer un message via un BMP devra tout d'abord déterminer la portée maximale de son message, tout comme il choisirait d'abord une adresse s'il voulait utiliser l'IP. Utiliser des distances mène toutefois à un besoin de connaître sa position géophysique de manière plus ou moins précise. Différentes méthodes pour connaître sa position ont été présentées lors de l'état de l'art et la partie II de cette thèse présente en détail la contribution du candidat sur le positionnement, IPADL. La sous-section 3.2 va plutôt s'attarder sur la notion de position en elle-même en plus de présenter comment le BMP en fait usage, sans se préoccuper de comment la position est obtenue. Ce chapitre prend aussi en hypothèse qu'il est possible pour tout objet d'obtenir sa position avec une précision suffisante pour effectuer son travail. Cette hypothèse est évaluée dans les chapitres traitant d'IPADL.

Chapitre 3 – Proposition du BMP

Le mode de propagation du BMP est inspiré de certaines situations de la vie courante. La technologie de communication des émetteurs-récepteurs radios (Walkie-Talkie) fonctionne déjà sur le principe d'une communication à tous sans réseau avec une portée maximale du message. En effet, selon la puissance de notre émetteur-récepteur et étant donné le modèle de dissipation des ondes électromagnétiques, passé une certaine distance du point d'émission (reliée à la force du signal), notre appareil ne pourra recevoir un message reçu. Les Walkies-Talkies sont habituellement vendus en précisant cette distance, typiquement de 2 à 5 km. À la différence du BMP, la portée des messages est déterminée par l'antenne de chaque appareil émetteur. Le premier message d'un échange sur Walkie-Talkie est typiquement de demander si notre interlocuteur est à l'écoute. Cette étape est nécessaire puisqu'il n'y a pas de réseau, donc aucun moyen de savoir si un autre appareil est disponible pour recevoir. Même alors, l'émetteur-récepteur continue de se croire seul, ce n'est que son opérateur humain qui se convainc du contraire. Une différence notable avec le BMP concerne toutefois la persistance des messages. Alors que dans le BMP ils restent en mémoire un certain temps, ils ne sont aucunement conservés sur les émetteurs-récepteurs. Il faut les consommer immédiatement à la réception.

Un second comportement humain qui sert d'inspiration au BMP est l'art. Que ce soit par la peinture, l'architecture, la sculpture ou la rédaction libre, l'art sous toutes ses formes a comme caractéristique de chercher à communiquer sans s'adresser à personne en particulier. Chaque œuvre cherche à transmettre un message, émotif ou littéral, mais ce n'est pas tous ceux qui vont voir l'œuvre qui seront réceptifs au message. Cette idée est reprise par le BMP dans le sens où tous les objets à proximité d'un émetteur vont recevoir le message sans pour autant tous se sentir concernés par son contenu.

3.2 Réseaux filaires

La description du BMP est indépendante du médium utilisé. Toutefois, le modèle sans réseau du BMP est plus difficilement réalisable lorsque les objets sont reliés entre eux par une connexion filaire. C'est d'autant plus vrai si les objets sont des ordinateurs plus puissants comme des ordinateurs de bureau ou des tablettes électroniques. Sur ce genre d'objet, il n'est pas raisonnable de penser empêcher l'ordinateur de participer dans les réseaux des autres protocoles utilisés par le poste. De plus, la majorité des systèmes d'exploitation comme Windows, MacOS, iOS et Android ne permettent pas au développeur de venir interférer simplement avec les couches protocolaires en place. Sur ces objets, le BMP est plutôt susceptible d'être exécuté par-dessus un TCP ou un UDP, puisque ce sont souvent les protocoles du plus bas niveau accessible. Ainsi, bien que le BMP ne contienne aucun mécanisme de maintien de réseau, il est fort probable que certaines implémentations utilisent des réseaux conventionnels pour leur communication, particulièrement sur des médiums filaires.

3.3 Portée des données

Comme mentionné précédemment, le BMP vise à limiter la portée physique des données afin, entre autres, de réduire la pollution des fréquences dans un environnement intelligent. L'élément crucial au mode de fonctionnement proposé est que chaque composant d'un réseau d'objets puisse connaître sa position avec suffisamment de précision pour évaluer les portées dites physiques, c'est-à-dire basées sur un calcul de distance euclidienne. La notion de position étant d'une grande importance pour le BMP, il est important de préciser ce qui est entendu par ce terme. Une position fait référence à un endroit géophysique mesurable tel qu'il est possible de mettre un point représentant cet endroit dans une carte à sa projection en deux dimensions. Une position est donc exprimée dans un système de coordonnées ayant un point d'origine.

La solution proposée n'entend pas imposer de système de référence en particulier. En effet, bien qu'il soit habituellement pertinent d'utiliser le système de coordonnées international de latitude et longitude, ce n'est pas toujours adapté. Dans un appartement intelligent, il est peut-être préférable de parler de pièces et d'étages plutôt que de degrés et d'altitude. De plus, il n'est pas toujours optimal de traiter les positions comme des endroits absolus. Dans certains contextes, des positions relatives à des points précis sont préférables. Le protocole de communication sera donc ouvert aux différents systèmes de coordonnées. De fait, le calcul des positions pourra être étendu par n'importe quel utilisateur au moyen d'une addition de module (*add-on*). Ainsi, dans la configuration du BMP, il faudra indiquer quel module de positionnement il faut utiliser. Il serait aussi possible d'utiliser l'un des champs d'options de l'en-tête présenté à la section 3.5. Il sera alors de la responsabilité de cet utilisateur de s'assurer que son module puisse à la fois générer des positions et les comprendre afin de pouvoir calculer une distance. Ce chapitre présente deux exemples de modules de distance, l'un basé sur la distance euclidienne et l'autre sur un système abstrait.

Ainsi, tous les émetteurs de données de l'environnement connaissent leur position selon le système choisi. Cette position est inscrite dans l'en-tête du message à échanger. Ensuite, la portée du message est calculée selon les besoins du système et ajoutée à l'en-tête. Tout objet qui reçoit ce message est responsable de calculer la distance entre sa position et la position contenue dans le message pour déterminer s'il est dans le rayon de validité de la donnée. S'il y est, il doit ensuite la réémettre. S'il n'est pas dans le rayon, il détruit simplement la donnée sans la réémettre, limitant ainsi sa propagation selon le mécanisme déjà décrit. En l'absence d'un module d'évaluation des distances, un message ne saurait être retransmis par cet objet et tous les messages reçus sont alors détruits à la réception. Un tel objet pourrait exister et ne servirait alors qu'à créer des messages. On pourrait ainsi imaginer qu'un thermomètre indique simplement sa mesure de la température sans avoir les capacités informatiques nécessaires pour faire plus.

Tableau 3.1

Modes de positionnement

Mode de positionnement	Mode d'expression de la distance	Module de calcul
Coordonnées géographiques (ex : 11 125N ; 174 523O)	En mètres (ex. : 15)	Distance euclidienne
Mot (ex. : salon)	En mots (ex. : maison)	Ontologie
Coordonnées personnalisées (ex. : A15)	En mètres (ex. : 5)	Module d'un tiers parti

Au-delà des systèmes de coordonnées, il est proposé d'offrir un mode libre dans lequel la position courante et la distance limite sont des chaînes de caractères. Utiliser des mots permet une plus grande flexibilité quant à la géométrie de la zone de validité pour un message. En effet, lorsque la portée est exprimée en mètre, la zone possible forme une sphère autour du point d'origine, ce qui n'est pas toujours souhaitable. Un capteur de température situé dans une pièce pourrait vouloir communiquer seulement avec cette pièce et pas avec la pièce voisine, par exemple. L'utilisation de mots ne devrait toutefois se faire que pour les sous-réseaux d'objets appartenant tous à la même application. En effet, les objets doivent être capables de reconnaître les mots valides afin de déterminer s'ils doivent retransmettre le message ou non. Ce mode permet des utilisations plus poussées de la notion de limite en rendant possible l'introduction d'une hiérarchie de noms. Il devient alors possible d'envoyer un message à une maison complète en plus de pouvoir le faire par pièce, pour autant que le système puisse déterminer que la maison englobe toutes les pièces. Puisqu'il est possible pour l'utilisateur d'incorporer son propre module de calcul de distance, les possibilités sont alors infinies. L'utilisation d'une ontologie peut même permettre de faire du raisonnement sur des mots inconnus a priori. Ce mode est inclus d'emblée par l'expression générale « portée physique », bien qu'il ne corresponde pas à un calcul de distance euclidienne propre à des positions géophysiques. Le tableau 3.1 présente un résumé associatif des différents modes de gestion exposés dans cette section.

Étant donné cette limitation physique de propagation, la seule façon dont tous les nœuds d'un réseau sont assurés de recevoir le même message est de mettre une distance de validité infinie, ce qui n'est pas l'idée ici. Bien entendu, la situation risque tout de même de se produire sur un petit ensemble d'objets rapprochés les uns des autres, dans un appartement intelligent par exemple. Pour les plus grands ensembles, le jeu des distances limites devrait rendre ces situations improbables. Cette particularité devrait faire en sorte que le système puisse grossir indéfiniment sans réel ralentissement, tant que la concentration d'objets connectés n'augmente pas trop et que les messages demeurent de courte portée. Cette hypothèse sera bien sûr à vérifier dans une expérimentation à grande échelle qui n'a pas été faite dans le cadre de cette thèse. En fait, il serait possible d'inclure une distance maximale de validité dans une configuration du protocole. Ainsi, tout nœud devrait rejeter un message dont la distance dépasse la distance maximale. De cette façon, il deviendrait impossible pour les objets possédant uniquement ce protocole de participer à une attaque informatique par déni de service distribuée, comme celle ayant touché les serveurs DNS de la compagnie Dyn en octobre 2016¹.

Pour l'ensemble du travail présenté dans cette thèse, il est supposé que les unités de mesure des différents éléments sont les unités du système international. Ainsi, les distances de cette section sont supposées en mètre et les temps de la section 3.3 sont supposés en seconde. Toutefois, différents modules de gestion des distances peuvent utiliser différentes unités, tant que tous les objets recevant des messages utilisent la même unité pour assurer un traitement uniforme.

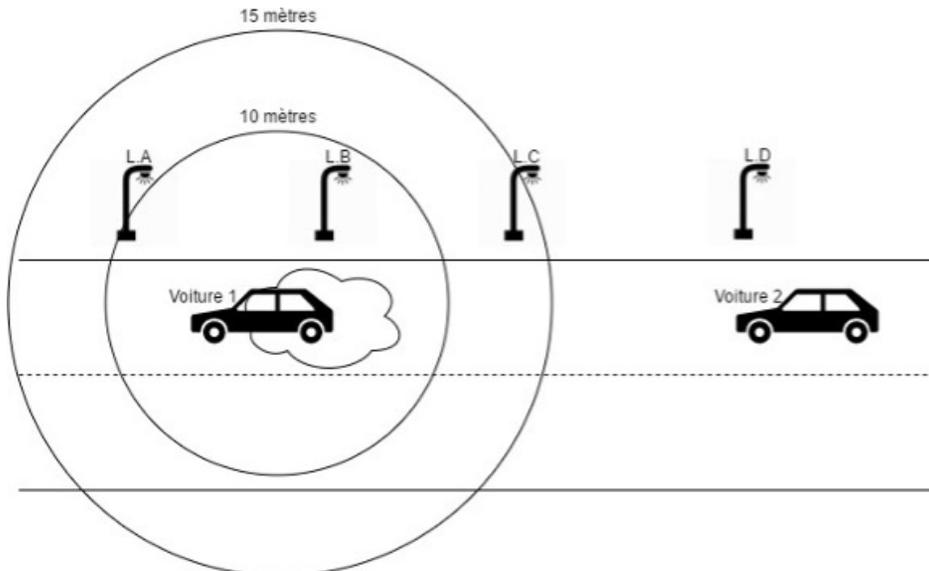
Examinons maintenant deux exemples d'application du BMP utilisant des distances physiques. La figure 3.1 présente un exemple d'utilisation possible du protocole. Il s'agit d'une situation typique dans les pays nordiques où une voiture circulant sur une route rencontre une plaque de glace. Dans un souci de coopération, la voiture décide de transmettre l'information pour informer les autres voitures dans le but d'empêcher les accidents.

¹ <https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/>

Dans cet exemple, l'antenne de la voiture peut émettre des messages sur une distance de dix mètres seulement. Or, la voiture 1, ayant rencontré la plaque de glace, souhaite transmettre l'information sur un rayon de 15 mètres. Par chance, le lampadaire B (identifié L.B) se trouve dans le rayon de dix mètres et reçoit le message. Ayant calculé qu'il se trouvait dans la zone permise, il retransmet le message. Les lampadaires A et C le reçoivent. À leur tour, ces lampadaires calculent la distance, acceptent le message puis le réémettent. Le lampadaire D reçoit alors ce message, mais comme il est hors de la zone, il refuse le message.

Figure 3.1

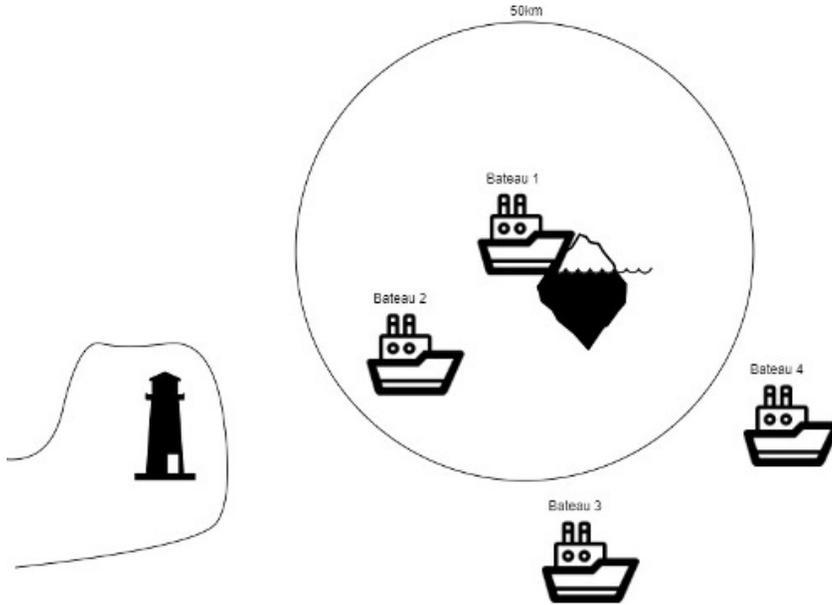
La voiture 1 rencontre une plaque de glace



Elle veut transmettre l'information dans un rayon de 15 mètres, mais ne peut émettre que sur dix mètres.

Figure 3.2

Un navire croise un iceberg et transmet l'information sur 50 km



La figure 3.2 présente une autre situation inspirée d'une histoire vraie centenaire se déroulant sur une échelle beaucoup plus grande que le premier exemple. Ici, nous avons le bateau 1 qui croise un iceberg sur sa route et décide d'avertir ses voisins au cas où l'un d'eux avancerait trop rapidement pour modifier sa direction à temps. Considérant que l'iceberg se déplace lui aussi, le navire décide de transmettre l'information dans un rayon de 50 km. Cette distance arbitraire reflète le niveau de confiance de l'opérateur du bateau quant à sa prédiction de la position de l'iceberg dans les prochaines heures, en tenant compte de divers éléments comme les vents, les courants marins et les itinéraires prévus des autres navires. Dans ce cas, seul le bateau 2 recevra l'information. Ainsi, les bateaux 3 et 4 de même que le phare côtier reçoivent le message, mais doivent l'ignorer puisqu'ils ne se trouvent pas dans la portée demandée.

3.4 Modulation de la puissance d'émission des capteurs

Dans le fonctionnement général du BMP, il est proposé de détruire les messages ayant dépassé la distance autorisée. Il y aurait toutefois un autre moyen de faire qui impliquerait moins de calcul pour les récepteurs : moduler la puissance d'émission pour faire en sorte que le message ne soit pas reçu au-delà de sa portée.

Les lois de la propagation des ondes électromagnétiques étant connues, il serait en effet possible de calculer la puissance de l'onde à émettre pour atteindre une distance donnée avec une force de signal suffisante pour la bonne réception de l'onde et du message transmis. Il serait même possible de précalculer ces puissances et de les inscrire directement dans une mémoire ROM pour ne pas exiger de calcul supplémentaire pour l'émetteur.

Cette solution serait sujette à être fortement limitée par les interférences présentes dans un environnement intelligent. N'importe quel obstacle pouvant réduire la force de l'onde réduirait d'autant sa portée et pourrait faire en sorte qu'un message ne se rende pas à un récepteur devant normalement recevoir ce message. Il serait donc nécessaire d'inclure une marge d'erreur dans le calcul de la puissance requise.

Puisque cette solution implique la construction d'une nouvelle famille de cartes réseau et de puces prenant en charge cette technologie, le concept ne sera pas poussé plus loin dans le cadre de cette thèse. Dans un scénario extérieur d'application du BMP, ce genre de technologie pourrait néanmoins être intéressant. Un exemple de scénario serait la détection et le contrôle d'oiseaux dommageables pour certaines cultures, tel qu'il a été envisagé dans cet article [53].

3.5 Gestion du temps

Dans la section précédente, l'aspect plus local du protocole a été énoncé. Il y est décrit comment limiter la portée des messages en se basant sur les positions des appareils en communication. En soit, le protocole serait fonctionnel tel quel, en permettant d'envoyer des messages dans un rayon donné. Toutefois, il ne répondrait qu'à une partie du mandat initial cherchant à réduire la quantité de données vivant dans l'environnement. Le BMP s'adresse avant tout aux systèmes d'IdO, dans lesquels les objets ayant des informations à communiquer sont majoritairement des capteurs. Les capteurs fournissent typiquement l'information qu'ils échantillonnent de l'environnement selon deux modes : à la demande d'un client ou par émission périodique par le capteur. Le BMP ne permettant pas d'envoyer un message à un objet précis, le mode à la demande n'est alors pas envisageable. Il faudrait qu'une demande unique lancée à la ronde soit comprise pour tous les capteurs. Le mode périodique est donc celui à considérer. Dans ce mode, une nouvelle lecture est disponible à une certaine fréquence. Cette nouvelle lecture écrase l'ancienne, la rendant obsolète. Plutôt que de laisser chaque application voulant utiliser cette lecture s'informer auprès du capteur de sa fréquence de lecture afin de déterminer lorsqu'elle pourra être oubliée, il est proposé que le protocole de communication s'en charge. À l'émission d'un message, un objet a la double responsabilité de choisir une portée physique et un temps de vie. Tenir compte de la fréquence de rafraîchissement des informations est une première motivation pour l'introduction d'une limite temporelle, mais ce n'est pas la seule. Une seconde motivation concerne l'espace mémoire limité des objets. Les microcontrôleurs Arduino sur lesquels ont eu lieu certaines expériences préliminaires ne disposaient que de 64 kilo-octets de mémoire au total. Il y a donc un nombre très limité de messages pouvant être contenus en mémoire à la fois. L'introduction d'une limite temporelle permet donc de libérer de l'espace plus facilement en fournissant un moyen simple de déterminer quels messages ne sont plus pertinents ou ont déjà reçu une mise à jour.

Une troisième motivation provient du caractère mobile de l'IdO. Bien que plusieurs objets soient fixes, de nombreux autres sont mobiles. En se déplaçant, ces objets vont rencontrer d'autres objets et échanger des messages avec eux. Sans durée de vie, ces messages, émis par un objet rencontré peut-être qu'une seule fois, resteraient en mémoire pour une durée indéterminée alors que l'objet les ayant émis n'est plus accessible. Dans un modèle de l'IdO où les données sont éphémères et sujettes à des mises à jour périodiques, il n'est alors pas normal de conserver un message ne pouvant être mis à jour. La limite temporelle prend en compte ces situations. La section 3.4.1 s'attarde plus en détail sur les objets mobiles.

L'idée de limitation temporelle des messages n'est pas nouvelle. Elle est déjà abondamment utilisée par le protocole IP pour empêcher qu'un message circule indéfiniment dans le réseau. La différence est qu'il est ici proposé d'utiliser une limite en temps réel, en secondes, plutôt que la limite relative induite d'IP, en nombre de réémissions (l'IP est en fait hybride : le temps est en seconde, mais chaque retransmission réduit le temps de 1. Étant donné les vitesses des réseaux actuels, il s'agit dans les faits d'un nombre de retransmissions et non d'un temps.).

Tout comme le protocole proposé détruit les messages ayant dépassé leur portée, il va aussi détruire les messages ayant vécu leur temps alloué de sa mémoire interne. Cette partie semble supposer l'existence d'un temps universel au sein du réseau de capteurs. Pourtant, selon la nature des données, cette limitation pourrait ne pas s'appliquer. En effet, comme expliqué dans la section précédente, les messages ne sont envoyés qu'aux voisins immédiats. Ainsi, pour la majorité des réseaux, la distance physique que le message va parcourir est très courte et la transmission sera donc quasi instantanée. Ce temps de transfert sera donc vraisemblablement négligeable, permettant le recours à des temps relatifs plutôt qu'absolus. Le temps de vie pourrait donc être raisonnablement exprimé en seconde à partir de la réception du message. Naturellement, il y a un ratio minimal à considérer entre la durée de vie de la donnée et le temps de transmission pour que cette approximation soit valide.

Ainsi, pour une durée de vie de 5 minutes, un temps de transfert de 300 ms ne représente que 0,1 % et est négligeable. Si la durée de vie n'était que de 5 secondes, le transfert de 300 ms représente alors 6 % du temps à vivre et peut ne pas être négligeable. Dans ces cas, un temps global est préférable. Ce sera à l'utilisateur de choisir la façon dont il souhaite gérer le temps. Le type d'objet sera alors aussi à considérer, puisque ce ne sont pas tous les objets qui mesurent le passage du temps avec la même précision.

3.6 Entreposage des données

Les sections précédentes ont exprimé les aspects touchant à la communication. Dans cette section, il sera question des aspects liés à l'entreposage des données dans les différents objets. L'objectif général est de faire en sorte que chaque composant du système n'ait accès qu'aux connaissances le concernant au moment opportun. Il s'agit donc, en quelque sorte, d'une base de connaissances distribuée. Pour y arriver, il est proposé d'entreposer les données dans un espace de tuples qui serait local à chaque objet. La caractéristique des espaces de tuple qui est la plus intéressante dans le cas présent est que les éléments y sont non modifiables. Pour modifier un élément, il faut impérativement le supprimer et en créer un nouveau avec la valeur modifiée. Dans le système proposé, le contenu des messages n'est pas modifiable, ce qui se prête bien à un espace de tuples.

Il est proposé que le protocole conserve l'original de chaque message reçu dans une mémoire interne. Ces messages sont conservés tant qu'ils sont valides, c'est-à-dire tant que la contrainte de temps et la contrainte de distance sont respectées. Pour des raisons d'efficacité énergétique et computationnelle, ces limites ne devraient pas être évaluées en continu, mais périodiquement. Le choix de la fréquence d'évaluation est laissé à l'utilisateur puisqu'elle dépendra nécessairement de la fréquence d'émission des messages et de leur vitesse de déplacement.

Une implémentation efficace calculerait le moment d'expiration du message selon son référentiel interne plutôt que de parcourir l'ensemble des messages à chaque seconde pour décrémenter le temps restant. L'implémentation Arduino, utilisée dans le prochain chapitre, fonctionne de cette façon. Elle utilise un dictionnaire trié où la clé est le temps local auquel le message doit être effacé.

3.6.1 Éléments mobiles

Les éléments mobiles posent de nombreux défis pour les protocoles de communication et le BMP n'en fait pas exception. En effet, selon la description actuelle, pour qu'un élément du réseau reçoive un message, il doit se trouver dans la zone de validité au moment de l'envoi du message, car il n'est jamais réémis. Ce mode de fonctionnement n'est pas intéressant pour les éléments mobiles. Si on prend l'exemple d'un usager portant un téléphone intelligent qui entre dans une pièce, il semble normal qu'il puisse accéder aux informations de cette pièce qui ont été émises il y a déjà un certain temps, mais qui sont toujours valides. La température d'une pièce est un bon exemple d'une donnée qui n'a pas besoin d'être mise à jour très souvent, mais qu'il est tout de même pertinent d'obtenir.

La question se pose donc : comment rendre les messages contenus dans les espaces de tuples locaux disponibles aux nouveaux éléments ? Une première façon consiste à ce que chaque élément réémette périodiquement ses messages, même si le contenu est le même. Ce n'est toutefois pas viable au niveau de l'efficacité énergétique puisque la majorité de ces messages seront inutiles si les éléments mobiles ne sont pas nombreux. Une seconde façon consiste à ce que l'élément mobile fasse des requêtes générales demandant une mise à jour des connaissances lorsqu'il se déplace. Naturellement, cette approche est plus énergivore pour l'élément mobile. C'est toutefois déjà la façon de faire pour de nombreux types de réseaux. De plus, les éléments mobiles sont habituellement munis de piles rechargeables sur place, contrairement à de nombreux capteurs qui nécessitent un remplacement complet de la pile.

Il est donc plus facile de recharger un élément mobile qu'un élément fixe, dans la plupart des cas réels envisagés. Toutefois, si les éléments fixes sont branchés sur le réseau électrique, cette considération disparaît. Il existe néanmoins un autre argument en faveur de cette méthode : celui de la pollution des ondes électromagnétiques. La première méthode engendre de nombreux messages inutiles qui ne font qu'augmenter les risques de collision ou de débordement dans les cartes réseau des différents éléments. La seconde méthode produit moins de messages inutiles, réduisant ces différents risques, sans toutefois les éliminer totalement. Les objets ne doivent cependant pas tous répondre en même temps, sinon le risque de collision est plus grand que dans le premier scénario.

Aucune des deux méthodes présentées n'offre une solution générale satisfaisante du double point de vue de la réduction des messages et de l'efficacité énergétique. Il est donc choisi de ne pas inclure de traitement particulier pour les objets mobiles directement dans le protocole et de laisser cette responsabilité à l'application l'utilisant, au besoin. D'un point de vue applicatif, il est en effet très simple d'analyser le contenu d'un message pour décider s'il s'agit d'une requête pour que l'application envoie, de nouveau, certains messages.

L'exemple présenté dans la figure 3.1 est un cas où il faut réfléchir au comportement des éléments mobiles. Dans la situation présentée, la voiture 2 se trouve à l'extérieur de la zone couverte par le message. Or, il s'agit d'un message pertinent que la voiture devra obtenir. Dans ce cas, ou le lampadaire C réémettra le message à intervalle régulier, ou la voiture demandera périodiquement à son entourage s'il y a des messages et les recevra quand elle entrera dans la portée de communication du lampadaire C. Cette situation illustre bien comment chaque solution génère des messages inutiles.

En conclusion, pour le BMP, la gestion des éléments mobiles se limite à recalculer périodiquement la règle de distance sur la mémoire interne. Ce recalcul devrait être effectué uniquement si l'objet s'est déplacé. Un seuil minimal de déplacement peut même être envisagé et être inclus dans la configuration du protocole pour réduire encore plus les calculs inutiles.

3.6.2 Temps et entreposage

Le protocole tel qu'il a été présenté jusqu'à maintenant présente une faiblesse en présence d'éléments mobiles. Il est mentionné qu'un message est conservé pendant un certain temps à l'écoulement duquel il est effacé. Il est aussi mentionné que le temps de transfert d'un message est considéré comme négligeable. Toutefois, il demeure possible qu'en présence d'éléments mobiles, le message soit retransmis pendant plus longtemps que sa durée de vie. Il serait alors reçu par un objet une seconde fois après que l'original a été supprimé, et il ne serait pas possible de le détecter. Ce serait aussi possible sur un rayon de validité suffisamment grand pour que les temps de transfert s'accumulent et ne soient plus négligeables. La situation peut se présenter en cas de saturation du réseau, lorsque les messages ne peuvent être réémis dès la réception, mais sont plutôt placés dans des files de réception et d'envoi.

La solution la plus simple serait de donner un instant de mort selon une horloge globale plutôt qu'un temps de vie relatif restant. Pouvoir donner un temps de mort pose toutefois l'hypothèse que tous les objets sont capables d'avoir accès à un temps global coordonné. Pour y arriver, il faut soit établir un temps universel par un protocole dédié à cette tâche ou utiliser une horloge externe. Ce n'est pas tous les objets qui ont accès à une horloge externe. Les algorithmes d'établissement de temps commun ne sont pas parfaits non plus. Concernant le BMP, le principal inconvénient des horloges globales est qu'elles nécessitent la présence d'un réseau dans lequel les objets sont connus. Sans cela, il ne saurait y avoir de temps coordonné commun à tous.

La seconde solution est d'inscrire le moment d'émission dans l'en-tête du protocole, en plus du temps restant. Naturellement, pour inscrire ce temps et l'interpréter, il faut avoir accès au temps commun et il vient d'être établi que tous ne pouvaient y avoir accès. Néanmoins, cette solution est préférable puisqu'elle demande seulement que ceux qui peuvent ajoutent le moment d'émission à l'en-tête et mettent à jour la durée de vie restante avant de réémettre le message.

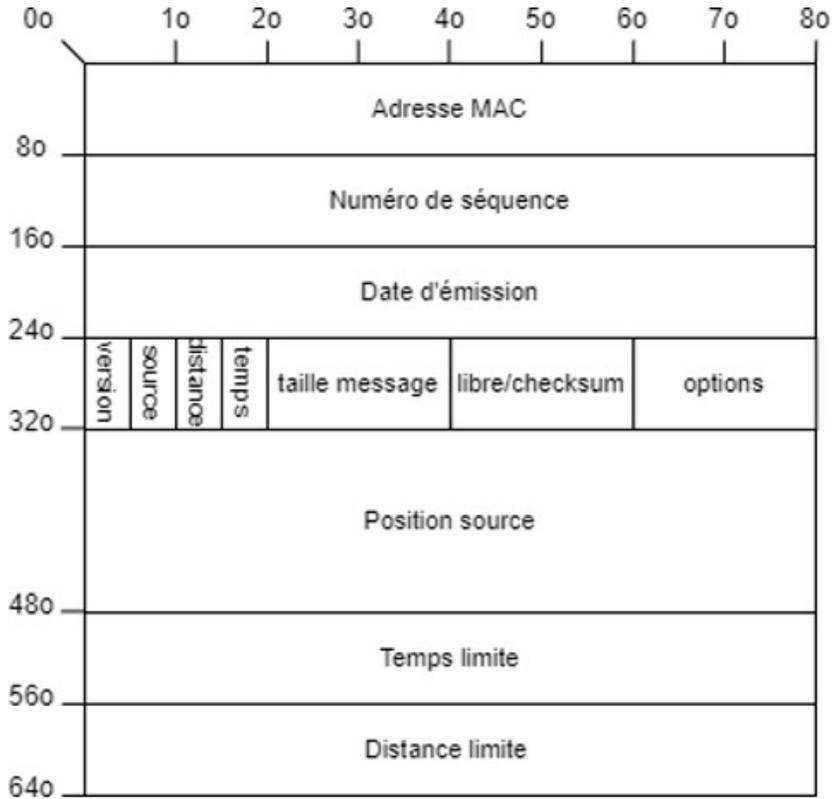
Le système n'est naturellement pas parfait, mais il semble préférable puisqu'il ne requiert pas la participation de tous pour être implémenté. Une hypothèse nécessaire à cette solution, de même qu'à la gestion du temps en général, est que tout objet est capable de mesurer le passage du temps avec régularité. C'est un service offert sur la majorité des architectures. Il doit toutefois être noté que la capacité de mesurer l'écoulement du temps est un élément nécessaire à l'utilisation du BMP, au même titre que la capacité à connaître sa position.

3.7 Format des messages

Dans cette dernière section de la présentation générale du protocole de communication, il sera question du format des messages. Plusieurs éléments sont à prendre en considération lors de l'élaboration d'un format de message, puisque celui-ci doit être fixe, mais tout de même flexible afin que son usage puisse évoluer. Un message contient généralement deux parties distinctes : un en-tête décrivant le message et un corps contenant le contenu du message. Alors que le corps du message peut avoir une taille arbitraire, l'en-tête est généralement de taille fixe ou calculable. La taille de l'en-tête d'un message BMP est de taille calculable, de 64 à 72 octets si la position est selon deux dimensions. Pour une version en trois dimensions, la taille est de 72 à 80 octets. Le format de l'en-tête doit refléter les besoins du protocole. Dans le cas présent, l'en-tête doit nécessairement contenir l'information sur la portée physique et la portée temporelle du message. Toutefois, il ne faut pas imposer un format précis pour les limites si on veut conserver la possibilité de les exprimer de manière absolue ou relative, comme expliqué à la section 3.2.

Figure 3.3

En-tête du protocole



Les marqueurs de taille sont en octets.

La figure 3.3 présente l'en-tête. Elle est alignée pour des architectures de 64 bits, soit huit octets. Les deux premières lignes forment un identifiant unique pour chaque message. Cet identifiant est composé d'un identifiant unique d'objet et d'un numéro de séquence monotone à débordement. L'identifiant unique d'objet est prévu pour être l'adresse MAC de la carte réseau de l'objet, mais n'importe quel numéro unique parmi l'ensemble des objets devant interagir fonctionne. L'utilisation de l'adresse MAC à 64 bits permet d'identifier de manière unique une carte réseau. En faisant la supposition que les objets connectés n'ont qu'une seule carte réseau, l'adresse MAC permet alors d'identifier de façon unique un objet. De plus, comme le protocole n'utilise pas les adresses IP, cela autorise l'objet à changer de sous-réseau sans pour autant perdre son identifiant unique. L'adresse MAC est généralement encodée sur 48 bits. Il existe toutefois des versions 64 bits qui devraient devenir de plus en plus populaires vu le nombre croissant d'appareils construits. Une adresse de 48 bits peut être simplement portée à 64 bits en ajoutant des zéros. La seconde partie de l'identifiant unique est un numéro de séquence pour le message. Ce numéro est unique pour chaque appareil et croît de façon monotone à chaque message envoyé. En utilisant un chiffre sur 64 bits, il semble invraisemblable que cet identifiant puisse atteindre la valeur maximale permise durant la durée de vie de l'objet. Si un tel cas se produit, la valeur déborde naturellement à zéro sans grand risque, tous les messages de cette période ayant normalement dépassé leur durée de vie. La combinaison de l'identifiant unique d'une carte réseau et d'un numéro unique de séquence pour cette carte forme alors une combinaison unique pour identifier un message.

La troisième ligne est celle qui indique le moment d'émission du message, lorsque connu. Le format proposé est d'utiliser le temps UTC en millisecondes. Toutefois, ce « temps » peut utiliser n'importe quel référentiel, pour autant qu'il soit commun à tous les objets devant interagir. Il a été établi que tous les objets n'auraient pas accès au temps commun. Dans ce cas, ce champ devrait rester vide. Dans l'éventualité où un objet dispose d'une horloge donnant le temps commun, alors cet objet doit remplir ce champ avant de retransmettre son message.

Avec le champ de temps de vie restant, il s'agit du seul champ modifiable de l'en-tête. Il est possible que deux objets remplissent ce champ en même temps, mais avec des temps légèrement différents. Dans ce cas, c'est toujours le temps le plus ancien qui est conservé. Il est pris pour acquis que cet objet était le plus près de point d'émission. Cela renforce l'importance d'avoir un temps commun cohérent. Un diagramme présentant comment ces valeurs sont utilisées est présenté à la figure 4.1.

La quatrième ligne de l'en-tête est celle donnant le plus d'informations. En partant de la gauche, les quatre premiers bits encodent le numéro de version du protocole. Il y a donc 16 versions possibles du protocole. Les 12 bits suivants encodent les formats utilisés pour la source du message, sa portée physique et sa portée temporelle, à raison de quatre bits par élément. Comme expliqué au cours de ce chapitre, il y a plus d'une façon de représenter ces éléments. Deux façons sont identifiées dans cette thèse, ce qui aurait pu tenir sur deux bits. Toutefois, il a été jugé plus approprié de se servir de l'espace disponible sur la ligne pour permettre à de nouveaux formats de s'ajouter. Les deux octets suivants renseignent sur la taille du message, en octet. La taille maximale d'un message serait donc de 65 535 octets, ou 64 kilo-octets. Il serait cependant optimal que l'utilisateur considère le type de réseau utilisé pour s'assurer que son message ne dépasse pas la taille limite d'un paquet. En UWB, par exemple, les messages sont de 124 octets utilisables au total (en-tête et corps). Les deux octets suivants sont laissés libres, pour un usage futur. Au besoin, ils peuvent être utilisés pour représenter un champ de contrôle de l'intégrité du message, sous la forme d'une somme de contrôle (*checksum*). Finalement, les deux derniers octets sont laissés libres pour permettre à l'utilisateur d'ajouter des options à son message, pour en faire l'usage qu'il voudra. Un exemple d'option serait d'y spécifier le format d'encodage interne au message. Un usage précédemment énoncé est d'indiquer quel module de calcul de distance utiliser, dans le cas où le champ de version ne suffit pas.

La section suivante encode la position de l'émetteur du message. Il s'agit d'une position fixe devant être connue. Il y a deux façons actuellement identifiées pour représenter cette position. La première façon est d'utiliser le système international de latitude et de longitude (et altitude). En encodant ces deux nombres sur 64 bits (soit habituellement un type double pour les langages typés), on obtient une précision à l'échelle du millimètre sur toute la surface de la Terre. Pour une plus grande précision, il est aussi possible de considérer qu'il faut seulement un octet pour encoder la partie entière d'une coordonnée et ainsi utiliser les deux octets restants pour la fraction donnant une bien plus grande précision. Cette première méthode occupe donc 2 lignes, ou 16 octets. Une version du protocole voulant gérer une troisième dimension d'une façon aussi précise que les deux autres pourraient simplement ajouter huit autres octets à la suite. L'analyse de la version du protocole permet de savoir la taille de l'espace réservé à la position dans l'entête. Les images et l'implémentation du prochain chapitre concerne la version en deux dimensions. Une deuxième façon de représenter la position est destinée aux systèmes relatifs, pour lesquels un simple nom suffit. En utilisant les mêmes 16 octets, il devient donc possible de donner des noms comportant jusqu'à 16 caractères ASCII.

À la suite de la description de la source, il y a la limite temporelle du message. Pour cette valeur, un seul mode est actuellement prévu pour la mise en production, occupant huit octets. Il est possible d'exprimer la valeur selon un temps restant exprimé en seconde. Cela suppose que le temps de transport du message est jugé nul ou négligeable. La partie entière du temps restant occupe les 32 premiers bits alors que la fraction occupe les 32 autres bits. Un second mode potentiel est celui où on donne une date limite basée sur le temps commun. Ce temps occupe aussi huit octets. Les cinq premiers sont réservés pour la date et l'heure tandis que les trois derniers servent aux fractions. Ces divisions dans les octets sont directement inspirées par la représentation interne des données dans le gestionnaire de bases de données MySQL. Ce mode n'est toutefois utilisable que si tous les objets ont accès à un temps commun.

À la dernière section, il y a le champ indiquant la distance maximale. Là encore, il y a deux modes à considérer. Le premier, occupant huit octets, est d'inscrire une distance réelle, en mètre. L'utilisation d'un double permet une précision actuellement jugée suffisante en permettant d'exprimer des distances à la fois très petites et très grandes. Le second mode est de donner un nom relatif, par exemple le nom d'une pièce où le message doit être pour être valide. Il est proposé d'utiliser 16 octets pour ce mode. Ainsi, selon le mode choisi, la taille du champ dans l'en-tête sera différente. Ce n'est toutefois pas problématique puisque cette information sera connue dès la troisième ligne de l'en-tête, qui jusque-là est fixe.

L'en-tête est ensuite suivi par le corps du message, d'une taille maximale de 64ko, comme mentionné précédemment. La taille maximale de l'en-tête est de 72 octets, pour un minimum de 64 octets pour la version 2D. Une troisième dimension ajoute huit octets à la position source et à la taille globale.

3.8 Retour sur les objectifs

Dans ce chapitre, un nouveau protocole de communication nommé *Bounded Message Protocol* a été introduit. Ce protocole répond au besoin d'avoir un protocole adapté pour la distribution de messages d'intérêt public locaux dans l'Internet des objets. Par sa structure, le BMP cherche à forcer l'élaboration d'applications distribuées pouvant fonctionner simplement à partir des données locales. Les faits saillants du protocole sont l'absence d'un réseau formel et la portée limitée des messages exprimées sous la forme d'une distance physique à laquelle s'ajoute une durée temporelle.

Les deux premiers objectifs de cette thèse ont été énoncés comme suit :

1. Concevoir un protocole de communication basé sur les positions des objets;
2. Concevoir un protocole de communication capable de gérer une durée de vie des messages.

Chapitre 3 – Proposition du BMP

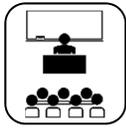
Considérant les éléments présentés dans ce chapitre, le candidat considère ces deux objectifs comme ayant été complètement atteints. Le mode de propagation du BMP est uniquement basé sur les positions des objets. Ces positions peuvent être exprimées de manière absolue face à un référentiel fixe ou encore d'une manière relative et abstraite. Ces positions servent ensuite à calculer des distances ou des appartenances, selon que les positions en jeu sont absolues ou abstraites.

Le protocole offre une gestion interne de l'entreposage des messages. Par le biais de cette gestion, la contrainte temporelle de la durée de vie des messages énoncée dans le second objectif est atteinte. Il est requis que l'émetteur d'un message renseigne à la fois de la distance maximale et du temps maximal de vie de son message. Le temps de vie est normalement inscrit en nombre de secondes restantes. Toutefois, si les objets possèdent un référentiel temporel commun, le temps de vie peut alors s'exprimer sous la forme d'un temps de mort. Pour éviter qu'un message dépasse sa durée de vie, son temps d'émission est aussi inscrit dans le message, lorsque c'est possible.

Les protocoles de communication sont habituellement agencés sous la forme d'une pile de protocoles où chacun apporte un bloc à l'édifice de la communication. La pile OSI est un exemple commun de la division des rôles des protocoles. On y définit d'abord des divisions pratiques pour conclure sur des divisions logiques. Au tout début, il y a la couche physique, soit la couche où la communication se produit par l'utilisation d'un médium physique. La couche suivante est la couche liaison, qui spécifie comment un objet doit communiquer avec un autre objet. Les couches suivantes sont plutôt logiques. Elles ajoutent des contraintes, des conventions et des abstractions pour faciliter l'acheminement et la compréhension des messages échangés. La place qu'occupe le BMP dans cette pile est délicate à définir. Assurément, il ne s'agit pas de la couche physique.

Au-delà, les principes à la base du BMP font qu'il est possible de le placer un peu n'importe où. Les prochains chapitres vont présenter une implémentation où le BMP se place au-dessus de la couche liaison dans une pile dont la couche physique est un UWB. Puis, une seconde implémentation sera présentée où le BMP se situe au-dessus de la couche réseau, dans une pile TCP/IP via Wi-Fi. En effet, le BMP demande seulement que les messages soient envoyés à tous, dans une diffusion. Si une couche inférieure désire conserver de l'information qui n'est pas nécessaire au BMP, cela ne le rend pas non fonctionnel. Il est toutefois prévu que la majorité des implémentations soient directement au-dessus de la couche physique, afin d'éviter toute communication superflue.

Ce chapitre s'est focalisé principalement sur la description théorique du protocole. Le chapitre 4 traite des problématiques liées à son implémentation. Le chapitre 5 détaillera les expérimentations qui ont permis de l'évaluer dans un environnement physique.



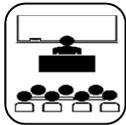
QUATRIÈME CHAPITRE

Implémentation



4.1 Caractéristiques communes

4.2 Différences entre les implémentations



4.3 Retour sur les objectifs

Dans le chapitre précédent, le protocole BMP a été présenté de manière conceptuelle. Ce chapitre va plutôt entrer dans les détails pratiques à considérer pour l'implémentation du protocole. Pour cette thèse, deux implémentations ont été réalisées, visant chacune un support matériel différent. Dès le départ, il a été prévu de réaliser deux implémentations dans le cadre de la thèse afin de mieux cerner les difficultés pouvant survenir. Tel qu'il a été énoncé dans le chapitre précédent, le protocole peut être utilisé à différentes couches logiques de la pile protocolaire OSI. Java et C++ ont donc été identifiés comme des langages en mesure de faire ressortir les éventuels défis de programmation et de déploiement. Le choix des plateformes d'intégration est aussi orienté vers ce but. En particulier, avoir deux plateformes permet de faire ressortir le défi d'hétérogénéité propre aux systèmes répartis.

La première implémentation du protocole a été réalisée en Java. La version utilisée est Java 8, bien que les fonctionnalités propres à cette version ne soient pas utilisées au niveau du protocole. Java est un langage offrant la particularité de pouvoir être exécuté sur un grand nombre de plateformes sans devoir modifier le code. Toutefois, cela vient au coût d'une machine virtuelle pouvant être gourmande en ressources. Pour cette raison, l'implémentation Java est orientée pour les appareils disposant d'une bonne puissance et d'une bonne capacité mémoire. Java étant l'un des langages pris en charge par le système Android, le mariage est parfait. L'implémentation vise donc les téléphones intelligents Android, tout en étant compatible avec les ordinateurs conventionnels.

Une implémentation naïve d'un programme en Java peut aussi exiger beaucoup de travail de la part du ramasse-miettes (« *garbage collector* ») et causer des ralentissements significatifs du programme. Il est donc important d'en tenir compte dans la gestion des objets Java en évitant au maximum d'en créer lorsque c'est possible.

Chapitre 4 – Implémentation

La seconde implémentation est en C++. Elle vise un déploiement sur un microcontrôleur DecaWino¹. Ce microcontrôleur est basé sur le Teensy 3.1/3.2 de PJRC et sur le module DWM1000 de DecaWave. Son matériel est plutôt limité, n'offrant que 64ko de mémoire vive et 72MHz de vitesse sur l'unité centrale de traitement. Ces contraintes nécessitent une implémentation efficace, autant en mémoire qu'en temps d'exécution.

La première partie de ce chapitre va présenter les caractéristiques communes (étapes de traitement, représentation interne et gestion des collisions) de l'implémentation Java visant les postes d'ordinateurs conventionnels et les téléphones intelligents Android et l'implémentation C++ visant les microcontrôleurs de type Arduino. La seconde partie présentera ensuite les différences dans la gestion du parallélisme. Une conclusion présentant comment ces implémentations remplissent les objectifs de la thèse terminera ce chapitre.

¹ <https://wino.cc/hardware/dekawino/>

4.1 Caractéristiques communes

Trois aspects seront exposés dans cette section sur les caractéristiques communes. Le premier concerne l'ordre de traitement des différentes contraintes du BMP. Le second aspect est celui de la représentation interne d'un message BMP. Finalement, le dernier aspect concerne la gestion du temps lors de la réémission de messages.

4.1.1 Étapes de traitement

La présentation du BMP dans le chapitre précédent a débuté par une explication du mode de propagation d'un message entre les différents objets impliqués. Par symétrie, il est naturel de commencer la présentation des implémentations par une description des étapes de traitement. La figure 4.1 présente un schéma de circulation d'un message. Prenons comme point de départ l'arrivée d'un message, c'est-à-dire le moment où il est entièrement disponible à la lecture par le programme. Les étapes complètes de réception d'un message dépendent grandement de la position du BMP dans la pile protocolaire et de la couche physique utilisée. Leur description dépasse le cadre de cette thèse.

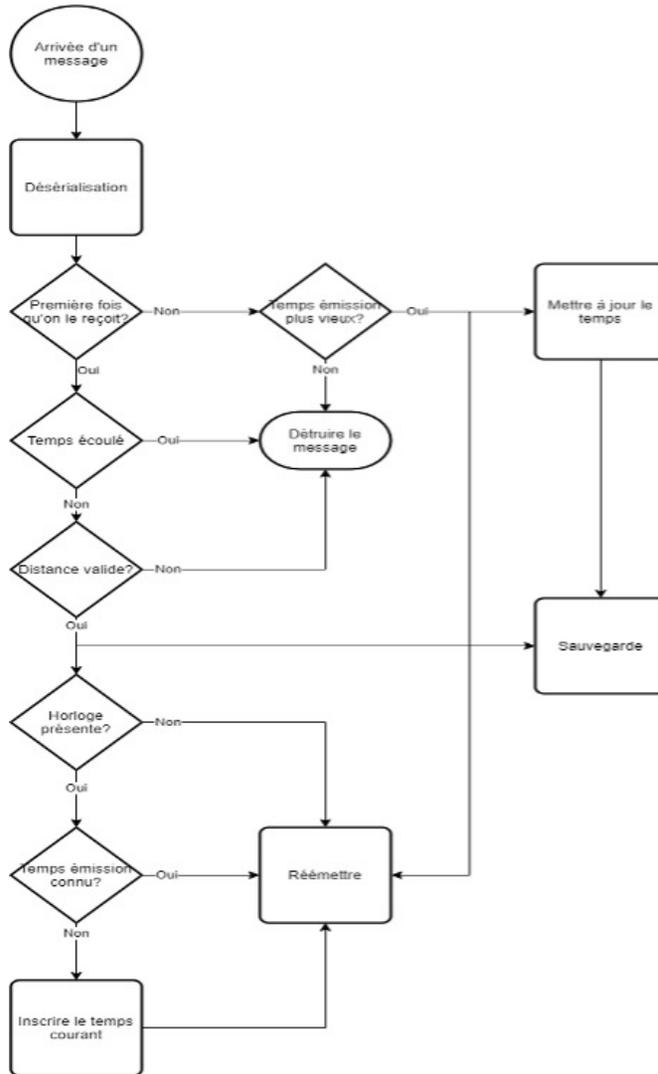
La première étape consiste à désérialiser le message reçu pour passer d'un flot binaire à un objet Java ou C++ où chaque champ de l'en-tête est accessible. La représentation interne exacte de cet objet est donnée plus loin à la section 4.1.2.

Du point de vue conceptuel, les deux étapes suivantes devraient être la vérification de la contrainte de temps et de la contrainte de distance. Toutefois, il a été jugé plus optimal de d'abord vérifier si le message a déjà été reçu dans le passé. Le BMP ne conserve qu'une seule copie de chaque message, les duplicatas peuvent donc être supprimés à la réception. Le mode de propagation fait aussi en sorte que les messages dupliqués sont nombreux, puisque le message est réémis par chaque objet le jugeant valide. Pour illustrer, imaginons un environnement comportant N objets. Un message est envoyé par l'objet 1, est reçu par les $N-1$ autres qui le jugent tous valide. Tous vont donc le réémettre. L'objet 1 va donc recevoir une copie de son propre message de la part des $N-1$ autres objets. Dans cet exemple, chaque objet reçoit donc chaque message $N-1$ fois. Chaque message ayant le même identifiant unique, il est facile de détecter les duplicatas et pertinent de le faire le plus tôt possible pour éviter des traitements supplémentaires. Étant donné le problème d'absence d'horloge globale énoncé à la section 3.4.2, cette première étape s'accompagne d'une seconde étape complémentaire pour la gestion du temps.

En effet, selon le traitement déjà décrit, il est possible de recevoir le même message plus d'une fois, mais avec des moments d'émission différents. Lors de la réception d'un duplicata, il faut donc vérifier si le temps d'émission est connu, et, si oui, s'il est plus ancien que celui actuellement dans la mémoire du protocole. Dans ce cas, il faut mettre à jour le temps pour conserver le plus ancien, considérer comme étant plus proche du point d'émission. Ensuite, il faut réémettre le message pour que les autres objets puissent mettre à jour le temps à leur tour. Advenant que le temps d'émission du nouveau message soit absent ou plus récent que celui en mémoire, le nouveau message est simplement effacé, ce qui met fin au traitement. Une implémentation visant un type d'objet n'ayant pas accès au temps global peut simplement ne pas programmer cette étape.

Figure 4.1

Étapes de traitement d'un message



Chapitre 4 – Implémentation

Les deux prochaines étapes sont celles d'évaluation du temps et de la distance. Les objets cibles de l'implémentation Java étant des téléphones intelligents, il est considéré que tous auront accès à une horloge globale. La vérification complémentaire de l'étape 1 est principalement présente pour un souci de complétude et de compatibilité entre les implémentations. Sur ces objets, il est jugé que l'accès à l'horloge est moins demandant en ressource et en temps d'exécution que l'accès au système de positionnement. La contrainte temporelle est donc vérifiée avant la contrainte positionnelle. L'idée est de réduire au minimum le nombre d'opérations pour avoir une implémentation la plus rapide possible. L'implémentation Java demande simplement au système d'exploitation le temps courant et le compare au temps inscrit dans l'en-tête du message, s'il y est. Si le temps n'y est pas ou encore s'il n'est pas écoulé, le message progresse à l'étape suivante. Si le temps est écoulé, le message est détruit, mettant fin au traitement. L'implémentation C++ considère qu'il n'y a pas d'horloge globale et ne vérifie donc pas ce champ.

La troisième étape consiste à vérifier la contrainte géophysique. Cette étape est de loin la plus complexe, étant donné les différents modes de représentation possibles. Dans une approche de type « boîte noire », l'étape se résume à donner la position d'émission et la limite à un module externe qui va répondre par un booléen. La réponse sera « Vrai » si la distance est valide, et « Faux » dans le cas contraire.

Dans sa plus simple expression, le module de positionnement consiste en un calcul de distance euclidienne en deux ou en trois dimensions entre la position d'origine et la position actuelle de l'objet préalablement calculée. Dans le cas d'un objet mobile, la dernière position connue n'est possiblement plus valide et alors il faut la recalculer. Android fournit des services permettant de maintenir à jour la position grâce au patron de conception « Observateur ». Un module permettant l'usage de position nommée est plus complexe. Dans sa version la plus simple, il confronte simplement le mot à une liste de mots autorisés. Il est toutefois probable que l'utilisateur voudra inclure du raisonnement sur les mots pour ajouter de la dynamique et de l'extensibilité au système.

Pour la première implémentation Java, seules les deux versions simples sont incluses, à savoir la distance euclidienne et la liste de mots valides. L'implémentation C++ suppose que la position est accessible via une variable. La position est soit renseignée lors de la configuration, soit reçue sur un port série. Lorsque le module de positionnement indique « Vrai », le message passe à l'étape suivante. Sinon, il est détruit, mettant fin au traitement.

La quatrième et dernière étape de processus de réception d'un message BMP est de l'ajouter en mémoire. Il s'agit simplement de copier le message dans un dictionnaire. La clé est le temps auquel le message devra être effacé, calculé à partir du temps courant de l'objet auquel on ajoute la durée de vie restante. En Java, un dictionnaire natif est utilisé alors qu'en C++ un dictionnaire sur mesure a été implémenté pour économiser de l'espace en n'implémentant que ce qui est utilisé. En Java, puisque les systèmes cibles sont habituellement bien fournis en mémoire, il est aussi possible de maintenir un second index sur le dictionnaire où la clé sera l'identifiant unique du message afin d'optimiser la première étape.

Les étapes suivantes du schéma de la figure 4.1 concernent le processus de réémission. La première étape consiste à vérifier que l'appareil dispose d'une horloge globale. L'absence d'horloge provoque l'envoi immédiat du message. Sa présence porte le message à la prochaine étape, qui consiste à vérifier si le temps d'émission est déjà connu (nous sommes dans un scénario de réémission). S'il est connu, le message est prêt pour l'envoi. Sinon, la dernière étape consiste à inscrire le temps d'émission comme étant le temps courant de l'objet selon son horloge globale. Alors finalement l'envoi a lieu.

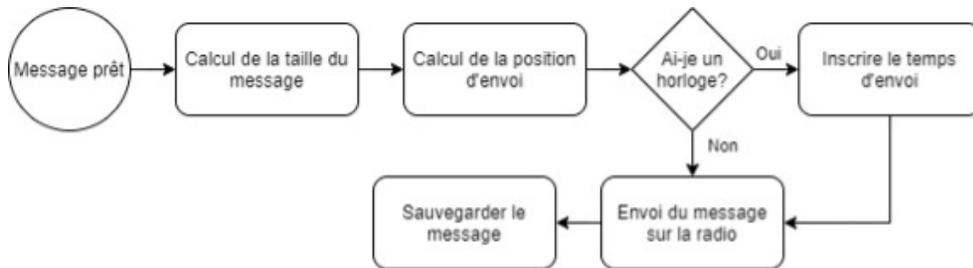
Tout comme les étapes complètes de réception d'un message sur les couches protocolaires inférieures dépassent le cadre de cette thèse, les étapes complètes d'envoi passées ce point seront omises pour les mêmes raisons.

Chapitre 4 – Implémentation

Le processus d'émission d'un message original est moins complexe. Il est illustré à la figure 4.2. L'utilisateur aura déjà rempli les champs de temps et de distance limite, en plus d'inscrire le contenu du message. Le protocole va donc tout d'abord comptabiliser la taille du message et l'inscrire dans le champ approprié. Puis, la position courante sera extraite d'une façon similaire au fonctionnement du module de positionnement utilisé lors du calcul de distance. Le module de calcul de distance fait directement appel au même module de positionnement utilisé ici. Puis, le temps courant est inscrit dans le champ approprié. Immédiatement après, le message est envoyé, de manière à ce que le temps d'envoi inscrit soit le plus près possible du temps d'envoi réel. La radio DWM1000 offre une fonctionnalité permettant de planifier le moment exact d'envoi d'un message. L'implémentation C++ utilise donc cette fonctionnalité pour renseigner ce champ de l'en-tête. Finalement, le message est ajouté à la mémoire interne. Il est en effet nécessaire que l'émetteur d'origine d'un message en conserve une copie dans sa mémoire afin qu'il puisse détecter les réémissions.

Figure 4.2

Étapes de l'envoi d'un message



4.1.2 Représentation interne

Le format d'en-tête défini plus tôt présente certaines difficultés au niveau de la représentation des données. Une première difficulté concerne la représentation du numéro de version et des formats pour les messages. Ces quatre champs sont, en effet, prévus pour une taille de 4 bits, permettant un total de 16 possibilités pour chaque champ. Or, sur une grande majorité d'architectures, il est uniquement possible d'allouer une adresse mémoire par segment de 8 bits. Les manipulations sur des tailles plus petites doivent se faire via des opérations arithmétiques ou des comparaisons logiques avec des masques. La deuxième difficulté est la gestion des champs de taille variable. Étant donné ces éléments, il a été choisi de ne pas utiliser le système de type de Java ni de C++ pour la représentation interne de l'en-tête d'un message, leur préférant un simple tableau d'octets natifs de la taille maximale. Sur ce tableau, chaque octet ou chaque séquence d'octets contigus est atteignable directement, ce qui simplifie les accès. Aussi, le champ renseignant sur la distance limite, qui est le seul de taille variable, occupe les derniers octets de l'en-tête. Cela a pour effet de rendre possible l'accès au premier octet de chaque champ via une constante définissant l'emplacement.

Lors de la création d'un message, un tableau de la taille maximale est réservé. Au moment de l'envoi, l'en-tête est alors recopié dans un nouveau tableau de la taille exacte requise, auquel est concaténé le corps du message, dont la taille exacte est aussi connue à ce moment. Lors de la réception d'un message, l'en-tête et le corps sont tous deux contenus du même tableau d'octets. La position du début du message est déterminée lorsque les quatre bits indiquant le format de la distance limite sont lus.

Naturellement, une enveloppe externe est fournie pour englober la représentation interne. C'est avec cette enveloppe que les utilisateurs ont à travailler. Elle donne un accès typé à chacun des champs, pour abstraire la représentation de l'usage. Puisque les champs représentant des positions et des distances peuvent être de types variables, une interface permet de les unifier, au cas où l'utilisateur préfère un usage générique.

4.1.3 Gestion des collisions

Les deux premières sections de ce chapitre touchent à des considérations d'implémentation ayant été réfléchies lors de la conception du protocole. Cette dernière section concerne plutôt un ajout dont la nécessité a été constatée lors des essais pratiques du protocole dans la plateforme de l'IUT de Blagnac. De par sa conception, le BMP suppose qu'il n'existe aucun réseau. Par extension, le candidat a supposé que le problème classique d'évitement des collisions dans un réseau ne se posait pas. L'hypothèse était que les messages ne seraient ni assez courants, ni assez volumineux, ni suffisamment importants pour qu'une collision éventuelle entre deux messages sur le médium pose problème. Bien que cette intuition n'ait pas été infirmée par l'expérimentation, le besoin d'une stratégie d'évitement des collisions est tout de même devenu évident au niveau de la retransmission des messages. En effet, puisque dans ces essais pratiques les objets étaient tous des DecaDuino exécutant le même programme, tous les messages étaient retransmis au même moment, ou presque. Les duplicatas ont été perdus, ce qui n'est pas très grave en soi, mais qui pose un défi pour la robustesse du système et la détection des doublons. De nombreuses solutions ont été proposées dans la littérature pour pallier ce problème [162, 18, 165, 48]. Toutefois, nombre de solutions supposent l'existence d'un réseau et d'une coopération entre les objets du réseau pour établir un horaire. D'autres stratégies supposent l'envoi d'accusés de réception pour confirmer la bonne réception d'un message. Ces solutions ne sont donc pas applicables au BMP. Ainsi, il a plutôt été choisi d'introduire un délai aléatoire avant l'émission de chaque message, un peu à la manière de Aloha. Le générateur de nombre aléatoire est initialisé avec le numéro de série de chaque objet, ce qui « garantit » des délais différents sur chaque objet. La valeur maximale du délai d'attente peut alors être choisie en fonction de la densité d'objets, de la fréquence des messages originaux devant être envoyés et de la tolérance à la latence du système.

4.2 Différences entre les implémentations

Les deux implémentations du BMP présentées dans ce chapitre ont été réalisées à des moments différents de la thèse et avec des objets cibles différents. Il s'en suit qu'il y a certaines différences entre les implémentations. L'implémentation Java, visant des objets plus puissants, utilise des outils de programmation qui ne sont pas disponibles sur les objets visés par l'implémentation C++. Plutôt que de lister toutes les différences entre les codes, cette section va plutôt s'attarder sur les méthodes de gestion du parallélisme.

Pour un fonctionnement optimal, le BMP a tout intérêt à utiliser la puissance du parallélisme, surtout sur les objets disposant de plus d'un CPU. Ainsi, il est recommandé de séparer l'obtention de la position, qui peut être longue, du reste du traitement. Android permet de le faire facilement à l'aide de fonctions de rappels qui vont modifier les variables internes du programme. De même, il est préférable de séparer le module de calcul de la distance dans un autre fil d'exécution, particulièrement si le module fait un raisonnement non trivial, comme interroger une ontologie. Ces étapes sont bloquantes pour le traitement d'un message et il peut être inutile de les séparer. Une étape qu'il est cependant utile de séparer est la réception des messages. Puisque le traitement des étapes présentées précédemment demande un certain temps, il importe de laisser le canal d'entrée libre de recevoir de nouveaux messages. L'implémentation du BMP peut alors prendre la forme d'un serveur de type « fil par requête », commun sur les serveurs web. Il faut toutefois noter qu'alors l'ordre de réception des messages n'est plus garanti. Le BMP ne garantit aucun ordre de message, mais une implémentation pourrait choisir d'offrir cette garantie. Un dernier endroit où le parallélisme est utile est le module de mémoire interne. En effet, ce module est chargé d'effacer les messages dès que leur temps de vie est écoulé. Mettre un module sur un fil d'exécution séparé lui permet d'effectuer son travail en toute indépendance des ralentissements possibles sur les autres sections du code, pourvu que le système d'exploitation lui accorde du temps de calcul.

Un lecteur familier avec le Teensy 3.1/3,2 aura déjà remarqué que cette puce n’offre pas l’accès aux fils d’exécution. Tout doit donc s’y produire de manière séquentielle. De plus, un programme pour cette puce, comme pour tout Arduino, n’a que deux fonctions dont l’appel est automatique. Lors de la mise sous tension, une fonction de *setup* est appelée. Cette fonction sert à initialiser les variables et les modules requis par le programme. Puis, il y a une fonction *loop* qui s’exécute dès que *setup* termine puis chaque fois qu’elle-même termine. C’est à l’intérieur de cette fonction que le cœur du BMP s’exécute. Or, cette fonction est toujours la même, on ne peut la changer. On peut simplement choisir quelles instructions elle exécute au moyen des structures de contrôle habituelles. Pour ces raisons, l’implémentation C++ du BMP repose fortement sur une machine à états. Cette machine rassemble toutes les étapes du schéma de la figure 4.1 et est programmée pour n’en exécuter qu’une seule par appel à *loop*. Le traitement complet d’un message prend donc en moyenne cinq cycles complets.

Autour de la machine à états, certaines étapes sont effectuées à chaque cycle. Il s’agit de la réception d’un message et la gestion du module de mémoire interne. La réception des messages consiste à vérifier sur le DWM1000 si un message est présent dans son tampon. Si c’est le cas, le message est désérialisé et mis dans une file d’attente qui sert de point d’entrée à la machine à états finis. Ce tampon permet de recevoir des messages sans devoir tous les traiter immédiatement. C’est particulièrement important puisque certaines applications de l’IdO vont déclencher des tempêtes de messages avant de redevenir silencieuses. C’est notamment le cas lorsque les objets doivent réagir à l’arrivée d’une personne dans un environnement. Cela permet aussi de recevoir les duplicatas des messages envoyés, ce qui sera utile pour mesurer la performance du protocole dans la prochaine section. En plus du tampon de la réception des messages, un tampon pour l’émission est aussi nécessaire. En effet, puisque les émissions ne se font qu’après un délai aléatoire borné, il n’est pas efficace de bloquer le traitement principal du BMP pendant ce délai. Le tampon d’envoi permet donc de traiter plus rapidement les messages entrant sans devoir ralentir pour envoyer les messages.

Étant donné que le BMP génère beaucoup de doublons causés par son processus de réémissions, il devrait donc y avoir plus de messages à recevoir que de messages à envoyer. Cette particularité est alors un argument suffisant pour l'introduction d'un tampon pour l'émission.

De même, le module de gestion de la mémoire interne s'exécute à chaque cycle. Il s'exécute immédiatement après le module de réception, mais avant la machine à états. L'objectif est d'effacer les messages le plus près possible du moment prévu. N'étant pas dans un système à temps réel, c'est le mieux qui peut être fait. Le faire au début d'un cycle assure aussi qu'aucune application devant s'exécuter après n'ait accès à un message périmé. Dans cette implémentation, les messages sont placés dans un dictionnaire fait main où la clé est le moment où le message doit être effacé. La navigation dans ce dictionnaire n'est permise que dans un sens et l'accès direct n'est pas possible.

En raison de la faible quantité de mémoire vive offerte par le Teensy, l'implémentation C++ pour Arduino est particulièrement à risque de dysfonctionnement dans un environnement dense en message. La gestion de la mémoire est donc importante sur cette architecture. Dans l'implémentation du BMP, il y a trois sections techniques à contrôler : le tampon de réception, le tampon d'envoi et le dictionnaire de stockage interne. Tous ces éléments ont une taille maximale fixe. Il est facile d'éviter les débordements sur le tampon d'envoi : il suffit de mettre la boucle principale de traitement en pause le temps que le tampon se libère. Si une mise en pause se produit, le tampon de réception devra absorber la charge des messages entrant le temps que l'envoi puisse reprendre. Pour le tampon de réception et le dictionnaire, il n'y a pas de moyen de contrôler les débordements autrement que par un refus pur et simple du message. Puisque le BMP est un protocole sans gestion de flux ni confirmation de réception, il n'est pas possible de notifier un interlocuteur de l'atteinte de la capacité de traitement, comme on pourrait le faire en TCP, par exemple.

Chapitre 4 – Implémentation

Ainsi, lorsque la mémoire interne ou le tampon d'entrée sont pleins, les messages sont perdus. Le tampon d'entrée lisant dans le tampon de l'antenne radio, un message entrant sera perdu au niveau de l'antenne. Le cas de la mémoire interne du BMP est plus complexe. Selon le moment où la vérification est faite, un message pourrait tout de même être reçu, vérifié et retransmis, même s'il ne peut être conservé localement. Dans ce cas, le même message pourrait être retransmis plusieurs fois, car l'objet n'en conserve pas la trace. Tel qu'il a été défini dans cette thèse, ce n'est pas le comportement souhaité du BMP. Afin de garantir l'invariant « *un message n'est retransmis qu'une seule fois* », il faut que l'implémentation s'assure de ne retransmettre que les messages vérifiés et enregistrés. L'implémentation Android est sujette aux mêmes risques de débordement des tampons, mais en raison de la plus grande capacité mémoire des objets Android, il est possible d'en augmenter la taille pour réduire le risque de débordement.

4.3 Retour sur les objectifs

Dans ce chapitre, l'ordre de traitement des étapes de réception et d'émission d'un message BMP a été précisé. Puis, certains détails concernant la représentation interne des messages et la gestion du parallélisme selon les implémentations ont été fournis. Ce chapitre s'inscrit dans la poursuite du premier objectif de la thèse, celui de la conception du BMP. Avec la présentation des détails d'implémentation, l'objectif 1 est complété. Les informations présentées dans ce chapitre permettent la réalisation d'une implémentation du BMP pour n'importe quelle plateforme. Les prochains chapitres présentent comment ces implémentations ont été utilisées sur de vrais objets afin d'évaluer le BMP.



CINQUIÈME CHAPITRE

Performance du BMP



5.1 Plateforme d'essai de Blagnac

5.2 Évaluations



5.3 Appartement intelligent du DOMUS

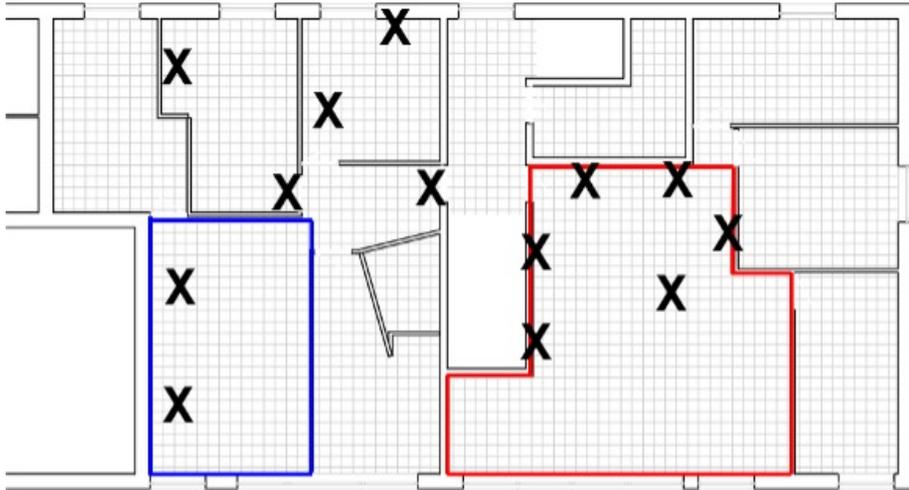
5.4 Retour sur les objectifs

La principale contribution de cette thèse est le protocole BMP, conçu pour permettre une communication libre d'informations générales dans un contexte d'Internet des objets. Un premier chapitre a présenté l'idée conceptuellement. Un second chapitre a précisé certains détails des deux implémentations (Java/Android et C++/DecaDuino). Pour ce chapitre, il sera maintenant question des expérimentations découlant de ces implémentations. Dans un premier temps, la plateforme d'essai de l'IUT de Blagnac sera décrite de même que les expériences y ayant eu lieu. Puis, dans un second temps, l'appartement du laboratoire DOMUS sera présenté, tout comme les expériences s'y étant déroulées.

5.1 Retour sur les objectifs

Dans le chapitre précédent, différents détails d'implémentation ont été présentés. La validité de l'implémentation C++ pour DecaDuino a été vérifiée expérimentalement dans la plateforme d'essai de l'Institut Universitaire de Technologie de Blagnac, en France. La plateforme est située dans un bâtiment de briques en Terracotta, à l'étage. Un plan de cet étage est présenté à la figure 5.1. La plateforme d'essai en elle-même est située dans la zone encadrée en rouge. Les croix indiquent de façon approximative la position des DecaDuino décrits dans l'introduction du chapitre 4.

Figure 5.1

Plan de la plateforme d'essai de l'IUT de Blagnac

Comme il est possible de le voir, la plateforme qui, à l'origine, était confinée dans la pièce en rouge a été étendue à l'ensemble de l'étage pour permettre la réalisation d'essais plus variés offrant le réalisme d'une maison comprenant plus d'une pièce. En plus des DecaDuino indiqués sur le plan, de nombreux autres objets communicants étaient présents dans l'environnement. Comme ils n'ont été utilisés pour aucune des expériences décrites dans la suite de ce chapitre, ils ont été omis du plan afin de ne pas inutilement l'alourdir. Pendant les expériences, il y avait dans toutes les pièces du plan d'autres objets électroniques diffusant des messages à des fréquences électromagnétiques variées.

Les DecaDuino étaient déployés de façon à ce qu'il ne soit pas nécessaire de les manipuler directement. Les DecaDuino sont connectés et alimentés par des Raspberry Pi via un fil USB vers MicroUSB, par groupe d'un maximum de quatre par Raspberry Pi. À leur tour, les Pies sont connectés et alimentés par un fil réseau les liant avec un serveur central. Un programme sur le serveur central permet de déployer automatiquement un programme sur les DecaDuino de notre choix via une commande dans une console. Les Raspberry Pi permettent aussi de communiquer avec les DecaDuino directement sur le port série. La communication entre les Raspberry Pi et n'importe quel poste sur le réseau interne du laboratoire se fait via *Message Queuing Telemetry Transport*, MQTT. Il est donc possible de lire en direct les messages envoyés sur le port série des DecaDuino sur un poste de travail. Le client MQTT Eclipse Mosquitto a été utilisé pendant les expérimentations. Les DecaDuino sont munis d'une radio UWB pour communiquer entre eux. Pendant les expérimentations, aucun autre appareil que ceux utilisés pour l'expérience ne communiquait par UWB. Il y avait toutefois de nombreux objets communiquant avec d'autres types de radios (Wi-Fi, par exemple).

Dans le contexte d'Arduino, on utilise le terme « radio » pour référer à la composante de communication installée sur le circuit imprimé. Cette puce contient habituellement une antenne pour la communication et le code de traitement de la couche physique. Selon le type de radio, le code contenu sur la piste peut aller jusqu'à inclure toute la pile protocolaire jusqu'à la couche transport. La radio UWB utilisé permettait de paramétrer la couche physique et de programmer à partir de la couche liaison du modèle OSI. Les puces Wi-Fi des téléphones Android, à l'opposé, ne donnent accès qu'aux protocoles UDP et TCP de la couche transport. Il faut aussi noter que UWB communique sur des fréquences différentes de Wi-Fi, rendant les antennes de chaque type de radio incompatibles entres elles.

Le serveur central enregistre aussi toutes les communications filaires sortant des DecaDuino pour en faire un registre. Il y a un registre par appareil, groupé dans un dossier par expérience. Pour l'ensemble des expériences, le registre indique quel objet a envoyé quel message, le contenu du message et s'il s'agit d'une retransmission. Il y est aussi indiqué chaque message reçu et s'il est accepté ou refusé. Étant donné l'espace mémoire limité des DecaDuino, il y avait un nombre maximal de messages pouvant être conservés. Le registre indique donc aussi si un message accepté n'a pu être sauvegardé ainsi que les messages qui ont été supprimés à la fin de leur vie prévue. Tous les objets utilisés étaient fixes et leur position a été inscrite dans le programme.

5.2 Évaluations

Les essais ayant eu lieu à l'IUT de Blagnac constituent les premiers déploiements du BMP. Ainsi, ils sont surtout de nature qualitative. Ces essais suivaient plusieurs objectifs :

1. Vérifier que les messages cessent d'être retransmis;
2. Vérifier qu'un message est transmis au-delà de la portée d'une radio UWB grâce aux retransmissions;
3. Vérifier qu'un message reçu hors de sa portée géophysique est effacé et qu'il n'est pas retransmis;
4. S'assurer que l'implémentation est assez rapide pour une utilisation en contexte réel de maison intelligente.
5. Puisque ces objectifs sont qualitatifs, les résultats qui en découlent le sont aussi. Des expériences quantitatives seront présentées à la section 5.2. Chaque objectif a été vérifié dans une expérience indépendante avec un sous-ensemble d'objets choisi pour favoriser une analyse efficace des résultats.

Pour évaluer le premier objectif, uniquement les DecaDuino de la pièce en rouge ont été utilisés. Un seul objet devait émettre un message alors que les autres étaient à l'écoute. L'analyse des résultats consistait alors à vérifier que chaque appareil avait reçu le message originel dans un premier temps, puis la réémission de chaque autre objet, une seule fois. Le premier essai a été un échec total. Si le message d'origine a bien été reçu de tous, ce n'est pas le cas des réémissions qui étaient absentes ou corrompues. En effet, à ce moment, aucun mécanisme d'évitement des collisions n'était prévu dans le protocole. Cette première expérience a donc permis de constater la nécessité d'en avoir un. À la suite de l'introduction d'un délai aléatoire (où le générateur de nombre aléatoire est initialisé par une graine différente sur chaque objet), l'essai suivant a donné le bon nombre de messages partout, permettant ainsi de vérifier l'objectif.

La configuration nécessaire à l'évaluation du second objectif est légèrement différente. En effet, les DecaDuino de la pièce en rouge sont tous en ligne de vue, c'est-à-dire que les radios UWB ont à la fois la puissance nécessaire pour atteindre chaque autre radio et qu'il n'y a aucun obstacle entre chacun. Une telle configuration ne permet donc pas de vérifier qu'un message dépasse la portée de la radio d'où il est émis la première fois. Ainsi, cette expérience a utilisé tous les DecaDuino situés en dehors de la pièce en rouge en plus de ceux situés sur le mur nord de cette pièce, selon l'orientation du plan de la figure 5.1. Aucun problème particulier n'est survenu et l'objectif a été rapidement vérifié en regardant dans les registres si tous les objets avaient reçu le message. Les mêmes registres ont aussi confirmé que les objets les plus lointains de la source (située dans la pièce en rouge) n'ont pas reçu le message envoyé de cet objet, mais bien d'un intermédiaire, prouvant qu'un message peut dépasser la portée de la radio pour atteindre des objets lointains grâce au mécanisme de retransmission.

La vérification de l'objectif 3 s'est faite au même moment que la vérification de l'objectif 2. En effet, la portée géophysique du message avait été calculée pour que tous les objets, sauf le plus lointain situé dans la pièce bleue, soient inclus dans la portée. Ayant déjà été établi lors de l'installation des objets par l'équipe de l'IUT que ceux de la pièce en bleu sont en ligne de vue, il a suffi de vérifier que les messages reçus sur l'objet lointain étaient refusés pour cause de distance dans le registre.

Le protocole expérimental pour l'objectif 4 est un peu plus complexe. Avant de le décrire, il convient de préciser ce qui est attendu d'une utilisation en contexte réel. Le contexte réel varie grandement selon le type d'application et la taille de l'environnement. La plateforme d'essai de l'IUT de Blagnac étant plus semblable à une maison intelligente qu'à une ville intelligente, il a été décidé de se concentrer sur le contexte réel d'une maison intelligente. Ainsi, dans cet environnement, certains capteurs vont périodiquement indiquer la valeur d'une mesure prise dans l'environnement. Ce genre de trafic est simple à prendre en charge et correspond au trafic des expériences précédentes. Le BMP prend donc en charge ce type de trafic. D'autres capteurs vont plutôt réagir aux actions des occupants (pensons à des détecteurs de mouvement ou des capteurs magnétiques de contact). Typiquement, les occupants ne sont pas si nombreux et alors leurs actions sont réparties dans le temps. Ce n'est toutefois pas toujours le cas. Une situation où les actions sont condensées est lors de l'arrivée des occupants sur place. À ce moment, un nombre plus important de capteurs vont s'activer en même temps (détecteurs de mouvement, luxmètres, thermomètres, détecteurs magnétiques de contact, etc.) et c'est donc pour ce genre de charge que le BMP doit être testé pour s'assurer qu'aucun message n'est perdu. Le protocole de l'expérience visant à vérifier l'objectif 4 consiste donc à avoir un grand nombre d'objets proches les uns des autres émettant des messages tous en même temps. On cherche donc à reproduire un vestibule au moment de l'arrivée des occupants. L'expérience se déroule en plusieurs phases.

À la première phase, un seul objet diffuse un message, mais à haute fréquence. Une période de 5ms a été choisie, bien que ce soit plus rapide que la période habituelle des objets domestiques. Pour chaque phase, on ajoute un objet émettant à la même fréquence. Qualitativement, les registres ont permis de constater que la majorité des messages originaux ont été reçus, mais pas toutes les réémissions. L'objectif n'est donc que partiellement vérifié. Pour être totalement vérifié, il aurait fallu pouvoir déterminer dans quelles conditions la performance du protocole n'est plus suffisante pour assurer une qualité de service minimale. En effet, bien que le BMP n'offre aucune garantie de livraison d'aucun message, il serait pertinent, pour le concepteur d'une application, de connaître dans quelles conditions il peut utiliser le BMP avec une certaine confiance que certains messages seront reçus. Ces conditions limites restent donc à établir. Les expériences permettent toutefois d'établir qu'une communication se produit dans les conditions énoncées, proches de celles d'une maison intelligente.

La plateforme d'essai de l'IUT de Blagnac a permis le premier déploiement d'objets utilisant le BMP. Ces objets sont des DecaDuino appartenant à la famille des processeurs Arduino et communiquent via une radio UWB. Les différentes expériences présentées dans cette section ont confirmé que le protocole est fonctionnel, en démontrant qu'un message atteindra tous les objets inclus dans la distance limite (à condition d'être à portée multi saut) et qu'il cessera sa progression au-delà de cette limite. La section suivante sur les expériences au DOMUS présentera des résultats plus quantitatifs sur le comportement du protocole dans des conditions légèrement plus exigeantes que les conditions réelles envisagées.

5.3 Appartement intelligent du DOMUS

Alors que la section 5.1 a permis de démontrer que le BMP a partiellement atteint l'objectif 4 de cette thèse, cette section vise à démontrer l'atteinte de l'objectif 5. Cet objectif s'articule comme suit : Déployer le protocole dans un scénario se rapprochant des conditions réelles.

Le BMP est un protocole général pouvant être utilisé dans une foule de domaines. L'expertise du laboratoire DOMUS se situant principalement dans les maisons intelligentes, ce cadre d'utilisation fut aussi choisi comme cadre pour un scénario se rapprochant des conditions réelles. Plus concrètement, le scénario consiste à simuler l'arrivée d'une personne dans une maison précédemment inoccupée. Comme expliqué dans la section précédente, ce scénario est l'un des plus exigeants pour une maison intelligente. On pourrait imaginer qu'un scénario de fête d'enfants ou encore d'incendie pourrait être plus exigeant encore, mais comme ces scénarios ne représentent pas la vie quotidienne, ils ont été écartés.

L'appartement intelligent du laboratoire DOMUS prend la forme d'un « vrai » appartement pouvant être qualifié d'un très grand trois et demie. Il se compose d'une chambre à coucher, d'une salle de bain, d'une cuisine, d'une salle à manger, d'un salon et d'un vestibule. Chacune de ces pièces est totalement fonctionnelle et totalement équipée. Au mobilier standard s'ajoute l'équipement électronique faisant de cet appartement un appartement intelligent. Sans faire la liste exhaustive, il y a des capteurs de mouvement dans toutes les pièces, des détecteurs de contact sur les portes des pièces et sur les portes des armoires, des thermomètres, des microphones et ainsi de suite. Le chapitre 7 va présenter plus en détail les capteurs RFID qu'on y retrouve aussi. La figure 5.2 présente un plan de l'appartement, en indiquant aussi la position des objets pour le scénario.

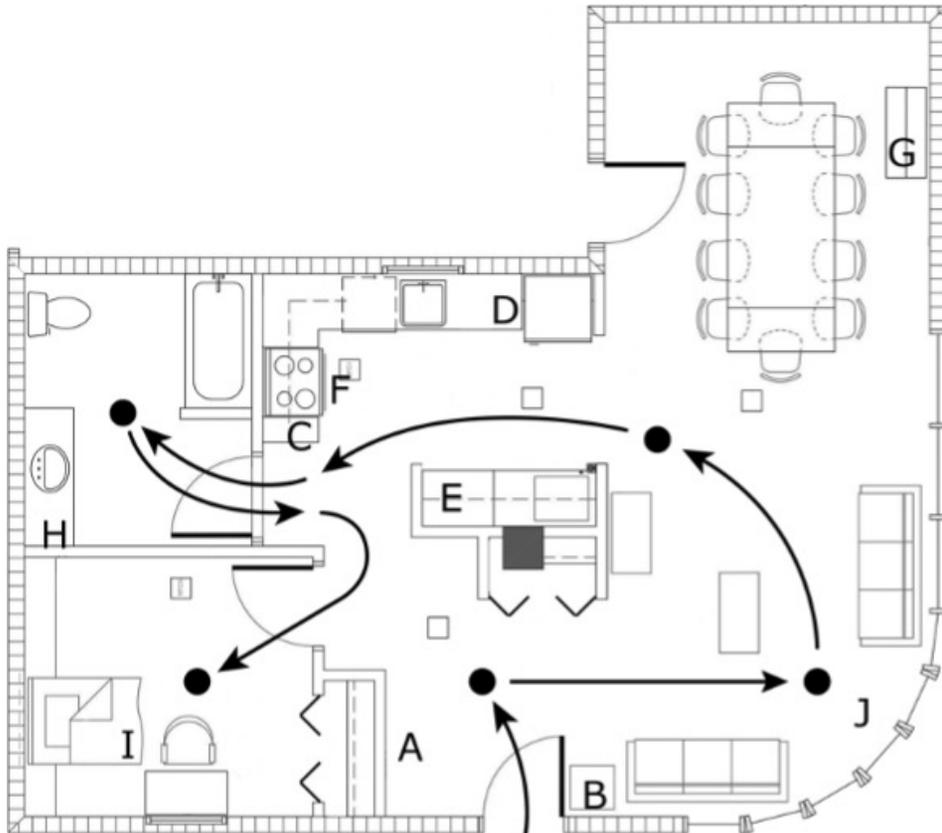
Chapitre 5 – Performance du BMP

Pour le scénario, il y a trois types d'objets. Il y a tout d'abord deux tablettes Nexus 7 fabriquées par ASUS exécutant la version 6.0.1 du système d'exploitation Android. Ensuite, il y a huit tablettes Nexus 10 fabriquées par Samsung exécutant la version 5.1.1 d'Android. Finalement, il y a un téléphone Nexus 5 fabriqué par LG exécutant la version 6.0.1 d'Android. Puisque tous ces objets tournent sur Android, le protocole BMP a été implémenté sous la forme d'un service Android utilisé par une application Android. Pour ce scénario, tous les objets sont connectés à un routeur Wi-Fi. Les messages du BMP s'échangent en utilisant UDP sur l'adresse de diffusion.

La configuration réseau exacte à l'intérieur du laboratoire DOMUS n'est pas simple à établir. Le routeur utilisé est un GL-X750 de la compagnie *GL Technologies (HK) Limited & Microuter Technologies Limited*. L'antenne Wi-Fi présente un gain de 2dBi sur la bande de fréquence 5GHz pour un taux de transfert théorique à 433 Mbps. La version ac (IEEE 802.11ac-2013) de Wi-Fi a été utilisée. L'UCT du routeur est un processeur Qualcomm QCA9531 cadencé à 650MHz. Plusieurs autres réseaux coexistent : le réseau général de l'Université de Sherbrooke, le réseau interne du laboratoire DOMUS et, dans certaines parties de l'appartement, un réseau Z-Wave. Les sources d'interférence sont donc nombreuses et il importe de conserver ces éléments en tête lors de l'analyse des résultats de l'expérience.

Figure 5.2

Position des objets dans le DOMUS et trajet de l'utilisateur



5.3.1 Scénario

Le laboratoire DOMUS est bien équipé en capteurs et en actuateurs, mais ne dispose pas vraiment des objets intelligents de la maison connectée. Les tablettes décrites précédemment vont jouer ce rôle. Elles ont été disposées dans l'appartement aux positions où on s'attendrait à voir des objets intelligents tels qu'un grille-pain, une cafetière, une lampe de chevet et ainsi de suite. Tel qu'il a été décrit dans l'introduction de la section, le scénario consiste à reproduire une situation dans laquelle une personne arrive dans l'appartement qui était précédemment inoccupé. La personne arrive donc au vestibule, va dans le salon un peu puis se dirige vers les toilettes. En chemin, elle s'arrête dans la cuisine pour un certain temps. À la fin, la personne va se coucher, mettant fin au scénario. Le téléphone joue le rôle du téléphone de la personne. La position exacte de ces objets est indiquée sur le plan de la figure 5.2. Les tablettes jouent les rôles suivants aux endroits suivants :

- A. Détecteur de mouvement dans le vestibule;
- B. Lampe de chevet sur la table près de la porte;
- C. Détecteur de mouvement dans la cuisine;
- D. Cafetière à gauche du réfrigérateur;
- E. Grille-pain face à l'évier de la cuisine;
- F. Cuisinière dans la cuisine;
- G. Horloge dans la salle à manger;
- H. Détecteur de mouvement dans la salle de bain;
- I. Détecteur de mouvement dans la chambre à coucher;
- J. Lampe de lecture dans le salon.

Dans la vie réelle, la majorité de ces objets vont émettre des messages qu'en certaines occasions précises et plutôt rares (lorsque l'eau bout, par exemple). Leur fréquence d'émission de message est normalement moins élevée que pour ce scénario. Pour faciliter la collecte des données d'utilisation dans le cadre du scénario, ces objets vont plutôt émettre plusieurs messages par minute, mais pas en même temps.

Le reste du scénario se déroule comme suit :

1. Au début, l'appartement est vide. Les objets D à G envoient quatre messages par minute jusqu'à la fin complète du scénario, en débutant après un court délai aléatoire;
2. Vers la 2^e minute, l'utilisateur entre et provoque des messages des objets A (mouvement) et B (lampe). L'objet A émet dix fois et l'objet B deux fois en une minute;
3. Vers la 3^e minute, l'utilisateur va vers le salon, activant l'objet J (lampe) qui envoie cinq messages en une minute;
4. Vers la 4^e minute, l'utilisateur va dans la cuisine, provoquant cinq messages en une minute par l'objet C (mouvement);
5. Vers la 5^e minute, l'utilisateur va dans la salle de bain, provoquant cinq messages en une minute par l'objet H (mouvement);
6. Vers la 6^e minute, l'utilisateur va dans la chambre à coucher, provoquant cinq messages en une minute de par l'objet I (mouvement);
7. Fin du scénario une fois la 6^e minute écoulée.

Dans ce scénario, le téléphone n'envoie aucun message. Il agit plutôt en tant que station d'écoute de référence, enregistrant tous les messages. Le trajet suivi par l'occupant pour déclencher les capteurs de mouvement est indiqué sur la figure 5.2. Pour ce scénario, le mode de communication utilisant des distances nommées est utilisé. Les messages sont envoyés soit à une pièce, soit à l'appartement. Les détecteurs de mouvement envoient des messages à l'appartement (sujet DOMUS), de même que l'horloge. Les objets de la cuisine n'envoient des messages qu'à la cuisine. Les lampes n'envoient des messages qu'à leur pièce respective (vestibule et salon). Chaque objet ne peut recevoir que les messages envoyés à sa pièce ou à DOMUS, à l'exception du téléphone qui reçoit tout. La vérification est faite en comparant la distance limite du message à une liste de noms connus.

Ainsi, dans cette configuration, tous les objets devraient recevoir la totalité des objets envoyés à la distance DOMUS. Les objets de chaque pièce doivent recevoir la totalité des messages émis pour chaque pièce. Chaque appareil est programmé pour envoyer un nombre exact de messages à la fréquence donnée par le scénario. Il est donc possible de calculer exactement le nombre de messages devant être reçus sur chaque objet. Les temps sont approximatifs puisqu'il s'agit d'une vraie personne qui se déplace dans le vrai appartement et non pas d'une simulation. Les résultats du scénario sont présentés dans la prochaine section.

5.3.2 Résultats et discussion

Le protocole BMP a été défini pour de nombreux cas d'utilisation. L'un de ces cas est le suivi des événements dans une maison intelligente. Afin de vérifier que le protocole fonctionne pour ce cas d'usage, un scénario a été proposé dans la section précédente. Cette section va maintenant présenter les résultats issus du scénario, de façon quantitative. Tout comme lors de la première phase d'essai à l'IUT de Blagnac, l'analyse se fera à partir de la trace laissée par les communications. L'application Android ici utilisée enregistre une trace pour chaque événement relié avec le protocole. L'analyse de ces traces permet donc de reconstruire exactement ce qui s'est produit. Pour l'analyse de cette section, les métriques qui nous intéressent sont :

1. Taux de messages originaux reçus;
2. Taux de messages réémis reçus;
3. Taux de messages refusés reçus;
4. Temps de traitement entre la réception d'un message et sa réémission.

Pour les métriques 1, 2 et 3, le taux visé est de 100 %. Un taux plus faible pour les métriques 2 et 3 serait toutefois acceptable puisqu'il ne s'agit que de réémissions et de message à refuser. Pour la 4^e métrique, le temps le plus court sera le mieux, car plus le traitement est rapide et plus la montée en charge est possible. Puisque les objets sont immobiles, il est possible de calculer exactement le nombre de messages devant être reçus par chacun (le dénominateur des taux 1, 2 et 3). Le taux 1 correspond donc au nombre de messages reçus ayant été accepté. Le taux 2 correspond au nombre de messages reçus dont la distance était valide, mais où le message était déjà en mémoire. Le taux 3 correspond au nombre de messages reçus dont la distance était invalide.

Chapitre 5 – Performance du BMP

Le scénario a été reproduit neuf fois. Dans chacun de ces neuf essais, les traces de l'objet D indiquent qu'il avait du mal à se maintenir sur le réseau. Tous les messages reliés à cet objet ont donc été retirés de l'analyse puisqu'il n'était pas possible de déterminer avec précision le moment où l'objet devenait indisponible. Pour la première expérience, l'objet G était mal configuré en acceptant les messages de la cuisine au lieu de ceux de la salle à manger. L'erreur a été détectée et réparée avant les essais suivants. Pour cette raison, les résultats du premier essai sont présentés dans un tableau séparé. Le tableau 5.1 présente ces résultats et les valeurs théoriques attendues.

Tableau 5.1

Résultats de la première itération du scénario

Objet	Envoyés	Recus (théorique)	Recus (trace)	% recus	Réémités	Doublons (théorique)	Doublons (trace)	% Doublons recus	Refus (théorique)	Refus (trace)	% Messages refusés	Temps moyen de rémission (ms)
A	10	42	39	93	39	454	387	85	260	240	92	215
B	2	50	45	90	45	454	365	80	260	229	88	335
C	5	95	90	95	90	650	572	88	16	15	94	243
D												
E	25	75	69	92	69	650	582	90	16	15	94	228
F	25	75	70	93	70	650	571	88	16	14	88	242
G	25	75	69	92	69	650	565	87	16	15	94	236
H	5	45	41	91	41	450	359	80	266	235	88	183
I	5	45	41	91	41	450	386	86	266	240	90	181
J	5	50	45	90	45	455	367	81	256	233	91	312
TEL	0	107	101	94	101	609	525	86	0	0	S/O	146

Les différentes colonnes du tableau sont les suivantes :

- Envoyés : Nombre de messages envoyés par l'objet, selon la trace;
- Reçus : Nombre de messages reçus par l'objet. Le nombre théorique est la somme des messages envoyés sur la distance DOMUS additionnée de la somme des messages envoyés sur la distance de la pièce de l'objet. Ces nombres sont utilisés pour le calcul de la première métrique;
- Réémis : Nombre de messages valides réémis par l'objet. Devrait être identique au nombre de messages reçus;
- Doublons : Nombre de messages valides reçus plus d'une fois. Le nombre théorique correspond à 9 fois le nombre de messages émis sur la distance DOMUS auquel on additionne le nombre d'objets dans la pièce multiplié par le nombre de messages émis sur la distance de la pièce. Il est à noter que dans une diffusion Wi-Fi l'émetteur du message en reçoit aussi une copie. Ces nombres sont utilisés pour le calcul de la seconde métrique;
- Refus : Nombre de messages pour lesquels la distance n'était pas valide. Le nombre théorique correspond au produit des messages envoyés à une autre pièce que celle où on se trouve multiplié par le nombre d'objets de cette pièce. Ces nombres sont utilisés pour le calcul de la troisième métrique;
- Temps moyen de réémission : Moyenne du temps écoulé entre la réception et la réémission de chaque message valide reçu. Il correspond à la quatrième métrique.

De ce premier tableau, on constate qu'entre 90 % et 95 % des messages valides sont reçus au moins une fois. C'est donc jusqu'à 10 % des messages qui ne sont jamais reçus selon les objets. Plusieurs raisons peuvent contribuer à ce résultat.

Chapitre 5 – Performance du BMP

La première raison est attribuable aux interférences électromagnétiques dans l'environnement. La puissance en un point donné d'un signal Wi-Fi n'est jamais stable dans le temps, ce qui peut provoquer des erreurs de réception. Le risque des collisions entre les messages est une deuxième raison qui demeure bien présente malgré l'insertion d'un délai aléatoire avant la réémission de chaque message. Pour cette expérience, ce délai était fixé à 50 ms, un délai qui semblait qualitativement bien fonctionner lors des essais à Blagnac. D'autres essais empiriques devraient être réalisés sur Android pour vérifier le choix de cette valeur. Il faut aussi rappeler que les conditions de cette expérience sont légèrement plus exigeantes que les conditions anticipées de la vie réelle. Le nombre d'objets de la vie réelle devrait dépasser rapidement le nombre de dix objets de cette expérience, mais leur fréquence d'émission pour la plupart devrait être bien inférieure à celles en jeu ici. Ainsi, un grille-pain pourrait n'émettre que lorsque son cycle de cuisson se termine, ce qui fait moins d'une dizaine de messages par jour, contrairement aux quatre par minute dans l'expérience.

La métrique 2 donne un résultat plus mitigé d'entre 80 % et 90 %. Ce résultat s'explique principalement par l'augmentation de l'utilisation du médium alors que tous les objets cherchent à communiquer dans la même fenêtre de temps. Comme le temps avant la réémission se situe généralement entre 200 et 300 ms, le délai de 50 ms n'est possiblement pas suffisant pour bien répartir les messages dans le temps.

Les résultats de la métrique 3 se situent entre ceux des deux autres métriques. C'était à prévoir puisqu'il y a moins de messages à refuser que de doublons. La congestion associée à ces messages est donc moins grande. Il y a aussi moins d'objets cherchant à réémettre un message.

Les temps moyens de réémissions incluent le délai aléatoire allant jusqu'à 50 ms avant la réémission. Malgré cela, ces temps semblent longs. Toutefois, il faut prendre en compte qu'Android n'est pas un système à temps réel et que plusieurs applications et services s'y exécutent en tout temps. Contrairement aux DecaDuino, le programme n'est pas le seul à s'exécuter sur Android et il doit partager l'unité centrale de traitement. Il faut aussi considérer que le programme est en Java et donc que le ramasse-miettes peut s'exécuter à n'importe quel moment, entraînant un ralentissement pouvant être marqué.

Le tableau 5.2 présente les résultats combinés des autres expériences, une fois l'objet G connaissant les bonnes distances auxquels il doit accepter les messages. Les résultats des trois premières métriques sont sensiblement les mêmes. On remarque toutefois que le téléphone a un taux de réception des doublons moins élevé que dans la première expérience. En analysant le taux par expérience, on constate que le taux est très variable d'une à l'autre, sans cause externe évidente. La cause la plus probable est qu'il s'agit d'un modèle plutôt ancien d'Android présentant des composants moins puissants que les tablettes. Il est donc possible que le téléphone ait eu plus de mal à maintenir la cadence à certaines reprises. Il serait intéressant de reprendre l'expérience avec du matériel plus moderne pour comparer les performances.

La constatation la plus intéressante à faire dans le tableau 5.2 concerne la quatrième métrique. Les temps de réémission des messages y sont globalement supérieurs à ceux du premier essai. Certains objets, dont le téléphone, prennent jusqu'à 500 ms avant de réémettre un message. Bien que l'application exécutant le BMP était la seule lancée par l'expérimentateur à être active sur les appareils, il est possible que les services internes d'Android se soient accumulés en cours d'expérimentation. Les appareils n'étaient pas redémarrés entre chaque itération du scénario. Il pourrait être intéressant de vérifier si le temps d'activité d'un appareil peut avoir un effet sur les performances.

Tableau 5.2

Résultats combinés des itérations 2 à 9 du scénario

Objet	Envoyés	Reçus (théorique)	Reçus (trace)	% reçus	Réémis	Doublons (théorique)	Doublons (trace)	% doublons reçus	Refus (théorique)	Refus (trace)	% Messages refusés	Temps moyen de réémission (ms)
A	80	336	308	92	308	3632	3070	85	1676	1524	91	293
B	16	400	359	90	359	3632	2914	80	1676	1487	89	535
C	40	759	721	95	720	4797	4214	88	128	121	95	395
D												
E	199	600	558	93	558	4797	4217	88	128	121	95	389
F	200	599	557	93	557	4797	4221	88	128	118	92	394
G	200	200	160	80	160	3600	3093	86	1724	1549	90	260
H	40	360	330	92	330	3600	2892	80	1724	1492	87	268
I	40	360	326	91	326	3600	2993	83	1724	1579	92	269
J	40	400	359	90	359	3640	2932	81	1644	1455	89	534
TEL	0	856	804	94	804	4869	3619	74	0	0	S/O	449

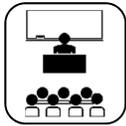
5.4 Retour sur les objectifs

Dans ce chapitre, deux expériences tenues l'une à IUT de Blagnac et l'autre à l'Université de Sherbrooke ont été présentées. La première expérience visait à vérifier que les règles internes au protocole provoquaient bien les comportements escomptés. Il s'agissait donc principalement de vérifier que la règle de distance empêche bel et bien le problème des inondations possible lors d'une communication *broadcast*. La seconde expérience visait à vérifier que dans les conditions plausibles d'un appartement intelligent le protocole est capable d'offrir une qualité de service acceptable.

Ensemble, ces deux expériences visaient à vérifier les objectifs 4 et 5 de la thèse. L'approche qualitative de la section 5.1 permet de vérifier partiellement l'objectif 4. Si l'expérience vérifie le mode de fonctionnement du protocole, elle ne permet pas de déterminer dans quelles conditions la qualité de service du BMP commence à se dégrader rapidement. De nouvelles expériences devront être menées pour établir ces conditions. Ces mêmes expériences devront avoir lieu pour les différents types d'environnement (domiciliaire, urbain, en nature, etc.) dans lesquels le BMP peut être amené à fonctionner.

L'approche quantitative de la section 5.2 permet de son côté de bien vérifier l'objectif 5. Un taux de réception variant entre 90 % et 95 % est jugé suffisant pour la bonne exécution d'une application d'assistance dans une maison intelligente. Ce taux cache toutefois le fait que l'exécution du BMP était lente sur les appareils utilisés avec un temps de réémission moyen des messages pouvant atteindre 500 ms. Bien que les traces de l'expérience de Blagnac ne permettent pas de l'affirmer, il a semblé à l'expérimentateur que l'implémentation C++ pour Arduino au BMP était plus rapide que l'implémentation Java pour Android. Pour les objets intelligents d'une maison, il semble plus probable au candidat qu'ils s'exécutent sur un microcontrôleur de type Arduino que de type Android. Dans ce contexte, les performances devraient être supérieures. Le principal argument est qu'alors le programme sera le seul en exécution et aura donc un accès continu à toute la puissance de calcul du microcontrôleur. Un environnement réseau plus exclusif devrait aussi permettre d'améliorer les performances du protocole. Le laboratoire DOMUS en particulier présente de nombreux réseaux en coexistence, ce qui n'est pas un environnement idéal pour évaluer un protocole de communication.

Ce chapitre conclut la présentation du BMP en tant que protocole. Dans le prochain chapitre, il sera question de l'utilisation à venir du BMP pour des applications réelles.



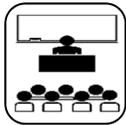
SIXIÈME CHAPITRE

Utilisations et travaux futurs



6.1 Aspects pratiques

6.2 Cas d'usage prévus



6.3 Conclusion

L'Internet des objets est l'un des domaines d'application de l'informatique présentant la plus forte croissance en ce début de XXI^e siècle. Les défis de ce domaine sont nombreux et cette thèse s'attarde à celui de la communication décentralisée. Le protocole BMP présenté dans les chapitres qui précèdent attaque ce sujet en proposant un modèle de communication dans lequel les messages qui s'échangent entre les objets sont géolocalisés et limités dans l'espace. Ainsi, les messages restent dans l'environnement immédiat des objets. Le BMP impose aussi une communication à tous, en ne permettant pas de choisir le destinataire d'un message. Ces caractéristiques impliquent certaines limitations inhérentes au protocole. Elles dirigent aussi le développeur à programmer certains types d'applications.

Ce chapitre est la conclusion de la partie de la thèse dédiée au protocole BMP. La seconde partie, présentée ensuite, présente en détail les contributions faites au domaine du positionnement intérieur via RFID afin de permettre une utilisation du BMP dans ces environnements. Dans ce chapitre, on formalise d'abord certains aspects pratiques jusqu'alors insinués seulement. Ces aspects pratiques incluent les conditions d'utilisation prévues du BMP, le type d'application cible et les limitations découlant de la conception du BMP. Puis, on présente certaines applications concrètes du BMP pouvant faire office de travaux futurs.

6.1 Aspects pratiques

Le sujet principal de cette thèse a été, jusqu'à présent, le protocole BMP. Le protocole a d'abord été présenté, puis son implémentation a été explicitée et finalement des essais pratiques visant à vérifier son fonctionnement ont été complétés. Tous ces aspects touchaient au protocole en lui-même. Cette section vise maintenant à présenter certains éléments concernant son utilisation pour de vraies applications. Trois éléments sont ainsi examinés : les conditions d'utilisation, les types d'applications cibles et finalement les limitations et garanties du protocole.

6.1.1 Conditions prévues d'utilisation

Un protocole de communication est toujours prévu pour répondre à un certain besoin déterminé par des conditions d'utilisation. Ainsi, MQTT¹ a d'abord été conçu pour connecter des *pipelines* à l'Internet en passant par des satellites. Le protocole devait être simple, rapide, peu énergivore et surtout résistant aux interruptions de service. Depuis, le protocole a évolué pour devenir une option attrayante pour de nombreuses applications IdO souhaitant rassembler les données sur un serveur central.

Le BMP a été conçu avec certaines des mêmes préoccupations en tête, à savoir que le protocole doit être simple, rapide et peu énergivore. Une différence notable est toutefois que le BMP ne permet pas de rassembler l'information ni de faire une communication dirigée d'un objet vers un autre. Le cadre d'utilisation envisagé pour le BMP est néanmoins bien différent de celui du MQTT. Le BMP s'oriente pour la communication locale d'information d'intérêt général alors que le MQTT s'axe sur la communication basée sur le principe de producteur-consommateur (*publish subscribe*) entre n'importe quels objets.

En effet, puisque le BMP n'inclut pas de mécanisme de découverte de pairs, il n'est pas possible de savoir quels autres objets sont disponibles pour communiquer. Tout message du BMP est ainsi envoyé à tous ceux qui peuvent le recevoir. Ainsi, n'importe quel objet présent à proximité de l'émetteur d'un message va le recevoir. Le contenu du message doit alors être d'intérêt public pour justifier ce comportement. L'idée est de partager du contenu factuel pertinent aux autres objets d'un environnement donné. Ce genre d'information est souvent produit par des capteurs de bas niveau et les messages sont alors des mesures sur l'environnement : présence de mouvement, température, débit d'eau, tension électrique, etc.

¹ <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>, consultée le 23 septembre 2020

Les limitations géophysiques et temporelles des messages BMP sont d'une importance primordiale au fonctionnement du protocole. Il est donc absolument essentiel que les objets l'utilisant soient capables de connaître leur position et soient capables de mesurer l'écoulement du temps sans trop dériver. La mesure du temps est habituellement possible sur les processeurs modernes, alors cet élément n'est pas trop restrictif. La capacité de connaître sa position n'est toutefois pas incluse par défaut sur les objets de l'IdO. Avant de construire une application utilisant un BMP, il faut donc d'abord s'assurer que les objets pourront connaître leur position. Si tous les objets sont fixes, ce problème peut se régler par une simple configuration. Pour les objets mobiles, une infrastructure extérieure au protocole sera sans doute nécessaire. De nombreux objets sont déjà compatibles avec GPS pour la localisation extérieure. Le système IPADL présenté dans le chapitre suivant est une alternative pour la localisation intérieure.

6.1.2 Applications cibles

Le point précédent a entrepris une conversation sur les conditions d'utilisation du BMP. La conversation se poursuit ici en portant cette fois l'attention sur les caractéristiques cibles des applications pouvant gagner à utiliser un BMP. La capacité à calculer sa position est la fonctionnalité la plus importante à avoir pour utiliser un BMP au point de vue protocolaire. Toutefois, dans une application, l'élément principal est de pouvoir fonctionner à partir d'informations partielles et sporadiques. En effet, le protocole ne garantit pas qu'un objet ayant reçu une information d'un autre objet reçoive sa mise à jour une fois la durée de vie écoulée. De même, il n'y a aucune garantie de recevoir un message. Le type d'application pouvant le mieux fonctionner avec ce genre de conditions est le système à base de règles. Dans ce type de système, l'application réagit aux informations reçues en fonction de règles préprogrammées. Il est possible, par exemple, de compter le nombre de fois qu'un ensemble d'événements se produisent puis de faire une action. Idéalement, l'application est aussi flexible sur la qualité du service, dans le sens où une donnée manquante n'a que peu d'impact. De nombreux exemples d'application à base de règles seront donnés dans la section 6.1 de ce chapitre.

6.1.3 Limitations et garanties du protocole

Le protocole proposé présente certaines limitations qui sont déjà visibles dans les autres chapitres de cette thèse. Cette section vise à présenter plus formellement ces limitations. Elles seront groupées en deux thèmes, soit les limitations physiques et les limitations informatiques.

Limitations physiques

Cette première section regroupe les limitations qui ont pour source les technologies en lien avec le transport de l'information et les objets en eux même. Tout d'abord, les spectres de fréquence utilisés se situent entre 700MHz et 5GHz pour les principales radios. À ces fréquences, les ondes électromagnétiques sont sujettes à des interférences lorsqu'elles traversent des matériaux métalliques, dont les murs et les humains. Il peut aussi y avoir des interférences entre différentes sources d'ondes, en particulier si elles émettent sur les mêmes fréquences. La seule amélioration possible sur ces points est qu'en limitant le nombre d'échange de messages, les risques de collisions d'ondes seront moins grands. Choisir un spectre de fréquence moins problématique n'est pas vraiment une option selon les règles du CRTC où seules quelques fréquences sont permises.

Un second aspect physique est la portée des ondes. Émettre une onde sur une grande distance requiert une grande quantité d'énergie, et cette énergie n'est pas toujours disponible sur les différents objets. Ainsi, il est possible qu'un objet n'ait pas l'énergie nécessaire pour émettre suffisamment loin pour couvrir la limite de distance souhaitée. Dans ce cas, si aucun objet n'est présent à proximité pour relayer le message, celui-ci sera perdu définitivement. Dans ce cas, un message ne va pas couvrir l'entièreté de sa zone géophysique souhaitée.

D'autres limitations physiques peuvent apparaître pour les petites valeurs ou pour les valeurs précises de distance ou de temps limites. En effet, les systèmes de positionnement actuels ont une précision variant de quelques centimètres et quelques millimètres. Il est donc impossible actuellement de vérifier une distance de l'ordre du micromètre. Ainsi, une application dans la micro-informatique est hors de portée, pour l'instant.

Du côté du temps, sauf pour les systèmes d'exploitation en temps réel, il existe toujours une certaine latence variable lors des envois de messages qui peut empêcher le système d'honorer des temps limites trop petits. Les échelles de temps dans les microsecondes et les nanosecondes sont donc à éviter en raison de cette instabilité. Naturellement, dans tous les cas, il faut aussi considérer les imprécisions inhérentes à tout instrument de mesure afin d'établir des marges d'erreur autour des valeurs. Le protocole présenté ne se soucie pas de ces imprécisions, ce sera aux applications à les prendre en considération. Le temps de transport est aussi considéré comme négligeable par le protocole. Sur une échelle de temps qui est courte, cette approximation n'est plus valide et le temps de transport devrait être prévu par l'application usant du BMP.

Limitations informatiques

Il existe aussi certaines limitations informatiques au protocole. Une première concerne les appareils mobiles. En effet, leur position risque de changer régulièrement et donc de devoir être recalculée périodiquement. Ce calcul demande du temps, surtout pour être précis. Or, en attendant de savoir sa nouvelle position, l'appareil ne pourra pas émettre de nouveaux messages ni vérifier la distance des messages reçus. Ou encore, l'appareil mobile devra faire ces opérations avec une position potentiellement inexacte en risquant de provoquer des erreurs logiques dans les applications.

Sur un autre point déjà mentionné en abondance, le protocole n'offre aucune garantie de livraison, pas plus qu'il ne permet l'envoi de messages à une destination précise. Ces éléments sont donc à considérer pour toutes les applications souhaitant en faire usage.

L'absence des mécanismes habituels de maintien de réseau rendent aussi impossible la mise en œuvre des algorithmes de synchronisation visant à établir un temps commun entre les objets. Si les objets souhaitent utiliser un référentiel commun pour le temps, il doit être établi à l'extérieur du BMP. Les options sont alors d'utiliser un autre protocole de communication ou encore d'utiliser une horloge externe. Sur certains types d'objets, aucune de ces solutions ne sera viable et alors l'objet n'aura simplement pas accès au temps commun. C'est pourquoi la durée de vie des messages est idéalement donnée en nombre de secondes après la réception du message. Même ainsi, il faut absolument que l'horloge du processeur de l'objet soit capable de mesurer le passage du temps avec un décalage minimal en comparaison avec la durée de vie des messages. Cette limitation rappelle l'hypothèse 3.

6.2 Cas d'usage prévus

La première section de ce chapitre prend la forme d'une discussion sur l'utilisabilité et les limitations du BMP. Le chapitre 5 nous amène vers quelques déploiements du protocole visant surtout à vérifier son fonctionnement. Cette section s'intéresse donc à certaines utilisations possibles en dehors du domaine expérimental. Tout d'abord, différentes applications seront présentées sommairement au tableau 6.1. Puis, deux applications seront explicitées avec plus de détails.

6.2.1 Assistance à l'errance nocturne

Au fil des années, le laboratoire DOMUS a travaillé à de nombreuses reprises sur des scénarios d'assistance technologique à l'errance nocturne que vivent des personnes atteintes de la maladie d'Alzheimer [94, 134, 133]. Le scénario consiste habituellement à détecter un mouvement inhabituel dans la chambre de la personne durant la nuit, puis à la guider vers les toilettes ou le salon et finalement à reconduire la personne vers son lit. La détection du réveil est faite à l'aide de détecteurs de mouvement dans la chambre et une plaque à pression détectant si quelqu'un est couché dans le lit. Le suivi de la position de la personne est fait à l'aide de détecteur de mouvement dans les différentes pièces et couloirs. Le guidage est fait au moyen de lumières qui s'allument devant la personne, que ce soit des lampes de chevet, des plafonniers ou encore des bandes DEL posées sur le plancher. Au besoin, une musique tranquille peut être jouée dans le salon pour calmer la personne avant de l'inciter à retourner se coucher à l'aide de signaux lumineux. Le scénario est variable selon la personne et ses besoins, mais l'idée générale est celle décrite précédemment.

Tableau 6.1

Exemples d'utilisation du BMP

Description	Contenu des messages	Distance limite	Temps limite
<i>Positionnement des transports en commun (bus, train, tramway, métro)</i>			
Le véhicule donne sa position. Chaque arrêt identifie d'abord si le véhicule y passe. Si oui, l'arrêt calcule le temps d'attente et l'affiche aux usagers.	-Position du véhicule -Identité du véhicule	Quelques kilomètres	Quelques minutes
<i>Prédiction de trafic</i>			
Les voitures indiquent leur destination. Des objets sur le bord de la route agrègent ces destinations et propagent les tendances. Progressivement, on arrive à savoir combien de véhicules se dirigent vers un endroit et quand ils y arriveront.	-Position du véhicule -Destination du véhicule	Quelques centaines de mètres	Quelques minutes
<i>Gestion d'un stationnement</i>			
Les voitures indiquent où elles se garent et quand elles libèrent un espace. Un contrôleur assemble l'information pour calculer le nombre de places disponibles et leur position, qu'il pourra indiquer à des panneaux informatifs	-Position du véhicule -Espaces disponibles	Taille du stationnement	Temps de stationnement prévu
<i>Adressage inversé</i>			
Bien qu'il ne soit pas possible d'envoyer un message à un destinataire, on peut s'imaginer une version légèrement différente où il est possible d'envoyer un message à une zone géophysique connue. Il suffit que la zone soit à portée d'antenne	Selon le besoin	Distance autour d'un point connu	Selon le besoin

Les éléments clés de ce scénario sont son mode de fonctionnement. Dans la description, on remarque que l'activation de certains capteurs déclenche les lumières. Il s'agit donc d'une application événementielle à base de règles, ce qui est exactement le type d'application souhaitée présentée à la section 6.1.2. Il serait donc possible de réécrire ce scénario en utilisant un BMP. Chaque capteur et chaque actuateur pourrait communiquer via le BMP pour s'échanger l'information. La lumière du couloir pourrait, par exemple, attendre de recevoir un message à la fois du détecteur de mouvement de la chambre et de celui du lit, voire d'un détecteur de contact sur la porte de la chambre avant de conclure qu'elle doit s'allumer. Le temps de vie des messages serait calibré pour ne pas allumer trop tôt au cas où la personne se recouche d'elle-même. Des détecteurs de mouvement suivant la position de la personne, chaque actuateur de chaque pièce pourrait savoir où est la personne et si c'est à son tour d'agir on fonction de la séquence des informations reçues.

Un désavantage du BMP pour cette situation est toutefois que chaque objet doit être programmé avec des règles différentes. Ces règles doivent aussi être personnalisées pour prendre en compte la configuration de l'environnement et les habitudes de la personne. Ce travail doit tout de même être fait dans une certaine mesure dans OpenHAB pour chaque déploiement des scénarios de Kenfack [94] et Radziszewski [134, 133].

6.2.2 Intersection intelligente

Le BMP est conçu pour fonctionner avec des applications qui ont seulement besoin d'information locale pour faire leur tâche. Le premier exemple prend place dans un environnement domestique où cette façon de faire apparaît comme naturelle puisque toute l'information de la maison est contenue dans la maison. Pour ce deuxième exemple d'application, l'environnement sera moins fixe. L'application de cet exemple est une intersection intelligente située dans une ville intelligente. Il s'agit de repenser le concept de l'intersection pour la ville du futur où tout est interconnecté.

Dans ce scénario futuriste, les infrastructures composées de feux de circulation et de panneaux de signalisation sont remplacées par de petites bornes aux coins de l'intersection pour l'usage des piétons. En effet, la signalisation à l'intention des véhicules est maintenant obsolète et est remplacée par un contrôleur sous la route capable de communiquer avec tous les véhicules pour les coordonner. Ce contrôleur utilise le BMP pour communiquer avec les véhicules, avec les bornes piétonnes et avec les signalisations restantes. Les objets participants à ce scénario sont les suivants : un contrôleur central à l'intersection, un annonceur en amont sur chaque route entrante, une borne par coin d'où un piéton peut traverser, un indicateur lumineux par route qu'un piéton peut traverser et finalement tous les véhicules circulant sur l'intersection. Chacun de ces objets communique avec le BMP à l'aide d'une application différente selon le rôle de l'objet. Les rôles et tâches de chaque catégorie d'objets sont :

- **Contrôleur central** : C'est le maître d'œuvre. Il recueille les intentions agrégées de chaque véhicule par les annonceurs et de chaque piéton pour optimiser la circulation, en accord avec les règles de gestion de l'autorité compétente. Il envoie des messages indiquant quelles voies sont ouvertes pour la période courante et lesquelles seront ouvertes à la période suivante. Une discussion sur la gestion des voies aura lieu juste après l'énumération des rôles. La durée de vie des messages correspond à la période de validité de la configuration incluse.
- **Annonceurs en amont** : Ils sont chargés d'indiquer aux véhicules arrivant la configuration de l'intersection. La configuration comprend les voies disponibles et les connexions vers les voies sortantes. Ils recueillent aussi les intentions des voitures pour les agréger avant de transmettre le résultat au contrôleur central. La durée de vie du message doit être suffisante pour que les véhicules aient le temps de traverser l'intersection.
- **Bornes piétonnes** : Elles indiquent la configuration de l'intersection et les voies actuellement ouvertes. Elles recueillent les intentions des piétons pour les transmettre au contrôleur central pour qu'il intègre les désirs des piétons dans sa prochaine planification. La durée de vie des messages est de quelques minutes, pour donner le temps au contrôleur de planifier les passages des piétons.
- **Indicateurs lumineux** : Lorsque le contrôleur envoie l'ouverture des voies, l'indicateur s'allume si sa voie est incluse dans les voies ouvertes, signifiant ainsi au piéton qu'il est temps de traverser. Ils n'envoient pas de messages.
- **Véhicules** : Lorsqu'ils reçoivent la configuration de l'intersection de l'annonceur, ils répondent avec leur destination, soit la voie sortante qu'ils souhaitent utiliser. La durée de vie de ce message est courte, puisque l'annonceur va traiter l'information dès la réception, ou presque.

Cette description correspond au système complet minimal. Il est possible de l'améliorer en ajoutant plus d'objets. On pourrait ainsi avoir un indicateur lumineux par voie directement sur la surface de la route pour indiquer de façon visuelle quelle voie est ouverte. Il est possible d'ajouter des systèmes de détection de danger ou de débris sur la route pour fermer des voies dynamiquement et avertir les autorités. On peut aussi imaginer que chaque piéton porte un objet renseignant sur son identité et sur sa forme physique pour ajuster le temps octroyé à la traversée, ou encore pour vérifier que la personne a fini de traverser avant d'ouvrir de nouvelles voies à la circulation des véhicules. Une autre amélioration intéressante est l'introduction d'un système de priorité pour le transport en commun et pour les véhicules d'urgence.

L'intersection de ce scénario suppose que tous les véhicules peuvent communiquer pour indiquer leur intention et recevoir des consignes. Seulement à cette condition, il devient possible de retirer les infrastructures de signalisation actuelles. Le scénario demande aussi que l'intersection se connaisse elle-même afin de pouvoir communiquer facilement sa configuration aux voitures.

La figure 6.1 présente une intersection en deux artères importantes. Les voies numérotées indiquent les voies entrantes alors que les lettres indiquent les voies sortantes. Une description de l'intersection peut alors se faire en indiquant une orientation, le nombre de rues impliquées, la quantité de voies entrantes et sortantes pour chaque rue et une matrice de connectivité. On peut aussi remplacer la matrice par une liste de connexion. Ainsi, on dirait que l'orientation de l'exemple est 0 degré, qu'il y a quatre rues, que la rue sud a deux voies sortantes et trois voies entrantes et que 0 connecte avec J, que 1 connecte avec G et que 2 connecte avec F et C pour les voies entrantes au sud. Voici une version plus synthétique telle qu'un annonceur pourrait envoyer à un véhicule :

Chapitre 6 – Utilisations et travaux futurs

Orientation : 0

Voies : [[2,3], [3,3], [2,3], [3,3]]

Connexions : [[0,J], [1,G], [2,F], [2,C], [3,B], [3,J],

[4,I], [5,H], [5,F], [6,E], [7,B], [8,A],

[8,H], [9,G], [9,E], [10,D], [11,C], [11,A]]

Les voies piétonnes peuvent s'exprimer en nommant les voies de chaque côté. Ainsi, la voie 8F représente la traverse piétonne de la rue nord. Une fois que la configuration est connue de tous, l'information sur les voies ouvertes correspond à une simple liste de connexions, comme dans l'exemple de la figure 6.1. Les voies ouvertes sont indiquées en vert sur l'image :

Véhicules : [[3,J], [4,I], [5,H], [6,E], [10,D], [11,C]]

Piétons : [2,A]

En indiquant l'orientation de la configuration, les véhicules, qui connaissent aussi leur orientation et leur direction, sont capables de faire la rotation du schéma qui s'impose afin de choisir la bonne voie permettant d'atteindre leur destination.

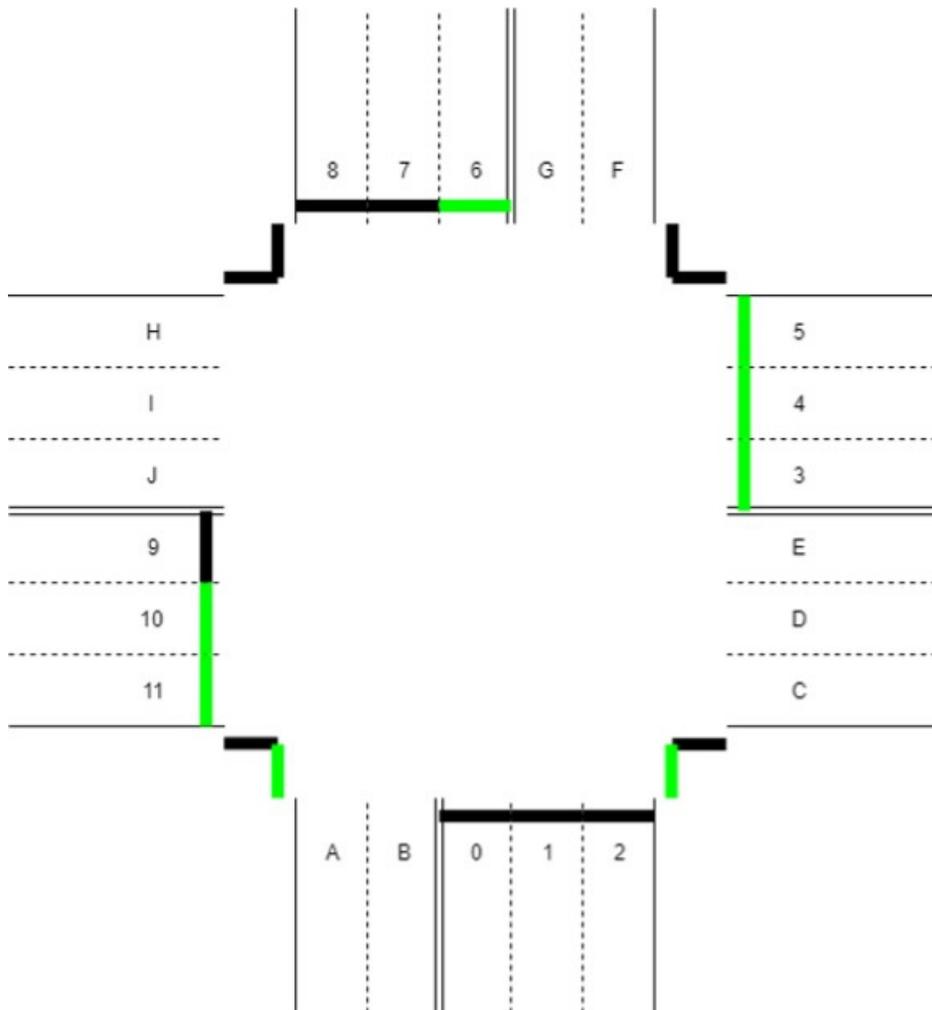
Les avantages d'une telle intersection sont multiples. Tout d'abord, il y a moins de matériel visuel, ce qui offre la possibilité de rendre l'intersection plus jolie. Ensuite, comme la voiture se charge de tout, il n'y a plus de danger d'inattention dans la conduite. De plus, l'environnement physique devient moins important. La signalisation ne peut plus être cachée par de la végétation ou de la neige et la visibilité routière n'est plus un facteur non plus. Il y a aussi d'immenses gains en efficacité qui peuvent être introduits par une meilleure synchronisation des voies ouvertes pour réduire le temps d'attente aux intersections.

Il y a toutefois une limitation à l'usage du BMP pour ce scénario. En effet, comme le BMP ne garantit pas la livraison des messages, il importe de se demander ce qui se produit dans ce contexte si un message n'est pas reçu. L'impact d'un message manquant est potentiellement plus grave que dans l'exemple de l'errance nocturne au sens où un accident provoqué par le protocole peut être mortel. Considérons les impacts objet par objet.

En ce qui concerne la communication entre le contrôleur et les bornes, voire avec les éventuels éléments lumineux, le risque de perdre un message est faible. Ces communications seront probablement filaires. Le fil étant peu utilisé dans ce cas, il devrait être fiable. Un message perdu entre une borne pour piéton et le contrôleur est sans danger, cela représente surtout un contretemps pour le piéton qui devra redemander un droit de passage. De même, un message vers l'indicateur de traverser pour piéton perdu ne fera que retarder son passage à sa prochaine demande. Ces situations sont frustrantes, mais pas dangereuses tant que le piéton ne décide pas de traverser sans attendre son signal.

Figure 6.1

Description d'une intersection



Les communications avec les véhicules peuvent conduire à des situations plus risquées. Il faut toutefois considérer que si les véhicules actuels sont capables de conduire dans notre monde actuel où rien n'est intelligent, ils ne devraient pas perdre cette capacité dans le futur. Le logiciel pilote sera donc capable d'improviser en analysant son entourage. Il devrait arriver à comprendre où sont les voies possibles et même qu'il devrait pouvoir détecter quand les autres véhicules se mettent en mouvement pour les imiter. Ainsi, la conséquence de ne pas recevoir la configuration de l'intersection de l'annonceur est minime, les systèmes d'observation de l'environnement de la voiture pourront pallier dans la plupart des cas. Sinon, rien n'empêche le véhicule d'envoyer un message demandant la configuration. Ce message pourra être répondu par tous les objets dans le rayon donné connaissant la configuration. Si la voiture n'est pas la seule à l'intersection, elle aura une réponse. Si elle est seule et qu'aucune réponse n'arrive, le risque encouru par une mauvaise évaluation du pilote est faible.

Ensuite, si l'intention envoyée du véhicule vers l'annonceur est perdue, le véhicule va seulement attendre qu'un autre véhicule ait la même intention avant que le contrôleur le prenne en considération. On peut aussi imaginer que le contrôleur aura accès à des détecteurs de présence pour déterminer s'il y a des voitures en attentes sur des voies où il ne devrait pas en avoir.

Finalement, si un véhicule ne reçoit pas son feu vert du contrôleur, il va simplement attendre et se retrouver dans la même situation que si son intention n'avait pas été reçue. Il risque aussi de se faire rappeler à l'ordre par un klaxon, si cette technologie existe toujours. Dans tous les cas, une redondance de l'annonceur permettrait une diminution des risques liés à un message perdu. Dans tout le cas, le principe de précaution appliqué en ingénierie automobile veut que la voiture soit capable de se débrouiller par elle-même dans toutes les situations possibles. Ainsi, l'application intelligente doit être vue davantage comme une optimisation du mode actuel de gestion que comme un remplacement du rôle du pilote d'un véhicule.

6.3 Conclusion

L'application de prédiction du trafic présentée dans le tableau 6.1 se jumelle bien à ce scénario d'intersection intelligente. En effet, ce scénario d'intersection demande à chaque véhicule d'indiquer la voie sortante qu'il compte prendre pour aider le contrôleur à planifier les voies ouvertes. Il serait simple de demander aux voitures d'indiquer leur destination à l'annonceur. L'annonceur serait alors responsable de compiler les intentions et de transmettre les tendances aux autres annonceurs pour former une image globale du trafic routier.

Ce chapitre conclut la partie de la thèse décrivant le protocole BMP. Ce chapitre s'est attardé sur l'utilisation future du BMP. Les applications souhaitant faire usage du BMP seront probablement des systèmes experts à base de règles. Ils doivent pouvoir fonctionner à partir de l'information reçue, sans avoir aucune garantie qu'une information reçue à un moment donné sera encore reçue dans le futur. Le BMP considère un temps de transfert des messages nuls. C'est une approximation qui ne saurait être vraie dans toute circonstance et donc les applications devraient en tenir compte au besoin. Certaines utilisations possibles du BMP ont été décrites plus en détail. Il y a tout d'abord un scénario d'assistance à domicile pour l'errance nocturne pouvant être implémenté dès à présent. Puis, il y a une intersection du futur permettant de rêver à la vie lorsque toutes les voitures seront autonomes.

La prochaine partie de la thèse se concentre sur les contributions faites dans le domaine du positionnement intérieur. La capacité de positionner un objet mobile dans un environnement intérieur est en effet essentielle à l'utilisation d'un BMP dans ce contexte.

TROISIÈME PARTIE

CONTRIBUTIONS EN POSITIONNEMENT

« - Et pourtant, ajouta Pencroff, qui montra une certaine difficulté à se résigner, le monde est bien savant !
- Quel gros livre, monsieur Cyrus, on ferait avec tout ce qu'on sait !
- Et quel plus gros livre encore avec tout ce qu'on ne sait pas, »
répondit Cyrus Smith. »

— Jules Verne, *L'île mystérieuse*, 1875

Cette thèse présente un nouveau protocole de communication basé sur les positions des agents l'utilisant pour communiquer. L'état de l'art, présenté au chapitre 2, a permis d'effleurer les nombreuses méthodes de positionnement d'objets exploitées en condition intérieure et/ou extérieure. Comme discuté, la précision et les conditions d'utilisation de ces méthodes sont très variables et il n'existe aucune approche universelle.

Les systèmes de positionnement par satellites sont possiblement ceux qui se rapprochent le plus d'un système universel pour les objets situés à l'extérieur des bâtiments. Toutefois, il existe de nombreuses situations où la précision chute grandement. Un exemple généralement cité est lors de la navigation dans un centre-ville [65]. Les gratte-ciels forment des écrans empêchant ou limitant les communications avec les satellites placés en orbite. Ces problèmes se règlent par l'ajout de stations terrestres de relais et par le croisement à d'autres sources de données de position, telles les antennes de communication cellulaire.

Au moment de débiter les travaux présentés dans cette partie, à l'hiver 2015, il n'existait pas de consensus au sein de la communauté pour une technologie de positionnement universelle pour positionner des objets à l'intérieur des bâtiments. Les systèmes à base de satellites ne sont pas vraiment une option due aux interférences subies par le signal en traversant les structures des bâtiments. Parmi les options technologiques présentant un bon potentiel à l'intérieur, on retrouve la RFID, le WiFi, l'UWB et le BLE. Les laboratoires DOMUS et LIARA auxquels appartiennent les directeurs de cette thèse ont choisi d'utiliser la RFID pour cette tâche, en raison notamment du faible coût des puces RFID passives. Ce sont donc sur la technologie RFID passive que les travaux de cette partie se concentrent.

Les usages de la position d'un objet sont multiples. L'usage le plus courant concerne la navigation. L'objet, généralement une voiture, est suivi en temps réel et des consignes de déplacement contextualisées sont envoyées à l'objet pour faciliter l'atteinte d'une destination. Les usages sont aussi nombreux dans le domaine de la construction, pour le positionnement des coins d'un terrain ou d'un immeuble, par exemple. Certains usages sont moins communs et s'adressent à un public plus spécialisé. La reconnaissance d'activités basée sur la position des objets est un bon exemple d'usage plus spécialisé [23].

En effet, bien que la maison intelligente du futur telle que présentée par la science-fiction occupe déjà l'imagination collective depuis un certain temps, les itérations actuelles s'adressent surtout aux personnes en perte d'autonomie. La clientèle cible inclut donc, mais sans s'y restreindre, des personnes ayant subi des traumatismes crâniens, des personnes souffrant de la maladie d'Alzheimer, et, dans une certaine mesure, les personnes âgées [3]. Le coût financier d'installation d'une maison intelligente est probablement l'élément le plus limitatif à l'adoption grand public. Les services offerts dans ces maisons sont multiples. On peut citer la détection de chute et l'assistance aux repas parmi les services courants. Cette clientèle étant habituellement sous la supervision de personnel médical, certains services de surveillance ont été développés pour aider le travail de ces intervenants. Un service important pour eux est la reconnaissance d'activités [3].

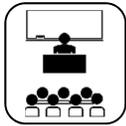
Il existe de nombreuses façons de faire de la reconnaissance d'activités. Il est par exemple possible d'utiliser des caméras vidéo pour voir directement ce que les gens font. Cette solution est toutefois considérée comme étant invasive et est donc surtout déployée dans les endroits publics. Les méthodes dites non invasives vont plutôt colliger des données provenant de sources indirectes pour inférer les actions des occupants de l'environnement intelligent. Ces données indirectes concernent généralement les objets utilisés, les ouvertures et fermetures des portes et des panneaux d'armoires et la position des occupants obtenue par des détecteurs infrarouges passifs. De nombreuses autres informations sont aussi disponibles.

Une information souvent négligée est toutefois la position des objets, et en particulier des objets ne nécessitant pas d'énergie pour fonctionner. Ainsi, le grille-pain est souvent inclus dans les modèles de reconnaissance d'activités par sa consommation électrique, mais pas la nappe pour la table à manger. En ne connaissant pas la position des objets, il est possible de savoir d'un occupant cuisine en utilisant la cuisinière, mais il n'est généralement pas possible de déterminer avec précision qu'il dresse la table. Une technique basée sur la position des objets, au contraire, sera capable d'identifier cette activité en suivant la position de la nappe et des couverts.

L'objectif des travaux présentés dans cette partie de la thèse est de concevoir un système de reconnaissance d'activités basée sur la position des objets. Cette position est obtenue en utilisant la technologie RFID passive. Les travaux concernant la reconnaissance d'activités de la vie quotidienne n'ont pas été menés à terme, le candidat ayant bifurqué vers la conception du protocole BMP avant la complétion du système. La contribution à la thèse de cette partie se situe donc seulement dans les travaux concernant le positionnement qui, eux, ont été menés à terme.

Toutefois, l'obtention de sa propre position est un besoin essentiel de tous objets désirant communiquer avec le BMP. Ces travaux concernant le positionnement intérieur via la technologie RFID passive sont donc directement reliés à cette thèse.

Cette partie de la thèse consiste en deux articles scientifiques. Le premier concerne l'élaboration de la méthode de positionnement des objets et son application au suivi de la position, le tout en temps réel. Le second article présente une amélioration à la technique rendue possible en considérant le signal RFID comme une série temporelle.



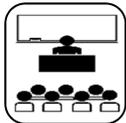
SEPTIÈME CHAPITRE

Tracking objects within a smart home



7.1 Introduction

7.2 Related work



7.3 Static positioning inside a smart home

7.4 Building models for the IPS

7.5 Dynamic tracking system

7.6 Experiments

7.7 Discussion

7.8 Conclusion

7.9 Acknowledgements

Résumé

Le BMP fonctionne sur l'hypothèse que chaque objet désirant communiquer est capable d'obtenir sa position. Dans le cadre d'un objet fixe, cette position peut être calculée lors de l'installation puis inscrite dans un fichier de configuration. Pour un objet mobile, la position doit pouvoir être calculée rapidement et dans n'importe quel type d'environnement. Les objets offrant aussi une puissance de calcul variable, mais généralement limitée, l'algorithme de positionnement doit être léger. Certains objets plus petits, comme les Arduino Uno, n'offrent qu'un très faible espace en mémoire d'exécution. Beaucoup d'objets basés sur Arduino n'offrent ainsi que 64 kilo-octets de mémoire. Il s'agit d'une pression supplémentaire sur l'algorithme de positionnement.

Dans l'article, le candidat propose une méthode de positionnement intérieur. La méthode est dite relative dans le sens où la position n'est pas une coordonnée, mais bien une zone de taille arbitraire. En modifiant la taille des zones, il devient alors possible de modifier l'efficacité du système. La méthode est générale et peut s'appliquer à n'importe quelle technologie permettant d'obtenir la force d'un signal entrant. La méthode ayant été développée au DOMUS, une application RFID passive est donnée. La première moitié de l'article présente la construction d'un jeu de données de position puis la sélection d'un algorithme de positionnement. Le critère principal du choix de l'algorithme est alors sa vitesse d'exécution, sans tenir compte de son coût en mémoire. La seconde moitié de l'article présente comment parvenir à établir la position d'un objet en déplacement.

L'article présente la méthode de positionnement de façon générale en visant une application en reconnaissance d'activités. La technologie RFID passive implique que la puce RFID ne dispose d'aucune interface programmable. Dans ce contexte, ce n'est donc pas l'objet lui-même qui peut calculer sa position. Pour une application directe au BMP, il faut donc un moyen permettant de transmettre la position en l'objet. Ce moyen est discuté plus en détail dans la section 3.2.

Commentaires

L'article a été publié dans le journal *Expert Systems with Applications* publié par Elsevier. Le candidat a effectué lui-même la majorité du travail. Toute l'implémentation et toutes les collectes de données sont le travail du candidat. Les autres auteurs ont surtout eu un rôle de conseillers pour améliorer la structure du texte. Kevin Bouchard a contribué à la réécriture de certaines sections à la suite d'une première version entièrement écrite par le candidat.

Cet article se base sur du matériel rédigé en 2016. En ce sens, l'analyse de coût du RFID face au Bluetooth n'est plus actuelle. Le matériel et le protocole Bluetooth ont bien évolué. Le matériel est devenu plus commun et plus abordable.

Cette thèse est au sujet du protocole de communication BMP. Pour fonctionner, le BMP exige que les objets en communication connaissent leur position dans l'environnement dans lequel ils sont. Cet article présente une méthode de positionnement pour objets dans un environnement intérieur, fournissant ainsi l'un des blocs de construction essentiel au protocole.

Source originale

Titre : « Tracking objects within a smart home »

Auteurs :

Frédéric Bergeron, Département d'informatique, Université de Sherbrooke,
Sherbrooke, Québec, Canada J1K 2R1, frederic.bergeron2@usherbrooke.ca

Kevin Bouchard, Département d'informatique et mathématique, Université du
Québec à Chicoutimi, Chicoutimi, Québec, Canada G7H 2B1,
kevin_bouchard@uqac.ca

Sebastien Gaboury, Département d'informatique et mathématique, Université du
Québec à Chicoutimi, Chicoutimi, Québec, Canada G7H 2B1,
sebastien_gaboury@uqac.ca

Sylvain Giroux, Département d'informatique, Université de Sherbrooke,
Sherbrooke, Québec, Canada J1K 2R1, sylvain.giroux@usherbrooke.ca

Keywords: RFID ; Smart home ; Data mining ; Decision trees ; Indoor tracking
system

Abstract

This paper presents a novel indoor tracking system built with common data mining techniques on radio frequency identification (RFID) tags readings. The system allows tracking of several objects in real-time in a smart home context and is a building block toward the deployment of an expert system to enable aging in place through technology. The indoor localization is modelled as a classification problem, instead of a regression problem as commonly seen in the literature. The paper is divided in two parts. The first one focuses on the ground truth collection that led to the model construction. The second part focuses on the filters that were designed to enable this model to be used in real-time in the smart home as a tracking software. Results from the first part show that most classifiers perform well on the static positioning of RFID tags task, with a random forest of 100 trees performing best at 97% accuracy and 0,974 F-Measure. However, collecting data to train the classifier is a long and tedious process. Results from the second part indicate that the accuracy of the random forest drops significantly when confronted with human interference. With the help of some filters, the tracking accuracy of objects can still be as high as 75%. Those results confirm that using passive RFID tags for an indoor tracking system is viable. Our system is easy to deploy and more flexible than trilateration or fingerprinting systems.

7.1 Introduction

In the recent years, technological advancements have led to the emergence of connected objects that can interact together and enable applications that would not have been possible before. Connected objects are commonly regrouped within a housing to form a smart home, which, in turn, enables many exciting possibilities [69, 33]. Western government are currently interested in using smart homes to enable the aging population to stay at home. For instance, smart homes can be seen as a way to secure the environment of the elders, or other persons, living with cognitive impairment to age-in-place, alone or with an informal caregiver [164]. This would reduce the pressure on health systems by delaying institutionalization, while improving the quality of life of the elderly.

To build this type of smart homes, it is required to install a wide variety of sensors and actuators interacting together to provide contextualized assistance (conscious intelligence) and implement basic safety rules (reflex intelligence). While it can be argued that the technologies to achieve a complete and functional smart home to enable aging-in-place are readily available, the artificial intelligence to build an expert system providing useful services is still inadequate. There are many factors explaining the flaws of current artificial intelligence in smart homes, but one of the most important is the low-level of information that can be extracted from distributed sensors. In this context, the sensors only provide raw data which must be transformed into high-level information in order to be useful for an expert system. In many cases, this task is straightforward. For instance, an electronic contact can assert if a door is opened or closed with a simple rule (if 0, then door_open). In fact, most non-invasive sensors only provide Boolean readings with time stamps. This apparent simplicity is, however, deceitful and results in many of the challenges faced by AI researcher in smart homes. Indeed, the recognition of the resident's activities is difficult to achieve by aggregating those low-level features [40]. This is why existing approaches can usually recognize high-level activities like *cooking dinner* or *dressing the table* but cannot tell what is being cooked or how many persons will share the meal [111, 88].

One major difficulty to improve the granularity of the activity recognition lies in the limitations of actual indoor tracking systems. Tracking systems are important in smart home as they facilitate activity recognition, which is in turn used by monitoring systems to predict danger or provide contextualized assistance [148]. There are two primary types of tracking systems: those that track inhabitants [36] and those that track objects [20]. For both types, many technologies have been proposed through the years. The most widely known is certainly the GPS and its competitors [107]. They rely on a certain number of satellites to compute the position of a receptor at any point on earth using time differentiation. They are commonly used in car navigation and tracking. While they are great for most outdoor positioning application, they lack in precision for indoor positioning. Moreover, the GPS signal is greatly affected by structures like walls and roof. In fact, GPS signal is often not available at all inside most building.

For indoor tracking of the residents, the well-known WI-FI [46] technology has been the most popular choice in recent years. It offers the advantage that most residential buildings nowadays are already equipped with one or more WI-FI routers. The main weakness of WI-FI is that the signal often vary due to numerous interference sources found in a house. When using the received signal strength indication (RSSI) for localization, there is also the multipath propagation problem that arises when a same signal can take different paths to reach a given tag. Another technology that is often used for indoor localization of residents in the same fashion as WI-FI is the RFID [82]. Depending on the band it is transmitting on, the signal power and the type of tags, its signal has a range of a few centimeters to several kilometers. The main weaknesses are also signal interference and multipath propagation. The false positive readings can also be problematic for some applications [113]. An important difference between WI-FI and RFID is the receptor, where passive RFID tags are much smaller and cheaper their WI-FI counterparts.

Other radio frequency technologies used for the task of indoor tracking include (but are not limited to) Bluetooth [132], NFC and ZigBee [135]. They all offer similar performances and limitations. NFC is mainly used for very short distance communication and for identification cards. It lacks range for continuous tracking, but is good to detect when a tag passes near a predetermined position. Bluetooth and ZigBee are both used for medium range communication. As tracking technology, they offer similar performances and limitations as RFID and WI-FI. One limitation of the Bluetooth technology comes from the pairing mechanism, where a master node can have at most 7 slave nodes.

Tableau 7.1

Comparison of radio frequency technologies for tracking

Technology	Range	Precision	Pros	Cons
Global positioning system	Earth	3-5 meters	Always available	Cannot be used inside
Wireless Fidelity	70 meters	Few centimetres	Already in most homes	Expensive tags
Passive RFID	30 meters	Few centimetres	Cheap tags	Expensive readers
Bluetooth 5.0	240 meters	Few centimetres	Low energy	Restricting pairing mechanism
ZigBee	100 meters	Few centimetres	Up to 65000 devices	Line-of-sight constraint

Table 7.1 resumes the main characteristics for the principal radio frequency technologies in use. The range column shows the range we can expect under normal circumstances, as advertised by their various developers ([43, 159, 106, 7]). The precision is left imprecise to better show the scale at which localization is performed [65, 161, 1, 116]. The aim of this paper is not to achieve the best precision but to show another, more stable and easier to use, approach to the problem.

The choice of technology often depends on the tracking context and on the objectives of the tracking. User's location is often used for context aware systems that adapt and personalize services to the specific person [44]. In that context, wearable technology such as a smart watch [105], a smart band [93] or a smart phone [60] are often coupled with the radio-frequency technology to gather the data. The main challenges in that case revolve around the energy consumption. Indeed, more data usually translate with more accuracy, but also into a higher transmission frequency resulting in a higher battery consumption on the wearable. It is also possible to use active tags, which are simply tags embedding their own internal battery power communicating with a fixed computer gathering the information (Bluetooth beacons [39], active RFID [47]). In that case, the wearable's battery limitation is eliminated, but there is a need to monitor the battery level of the active tags and maintain the system over time.

Due to those constraints, non-RF technologies are also in use. Systems based on technologies other than tags include ultrasonic sensors and microphones. Still, they have other weaknesses to consider. They can be misled by pets or even wind, for instance. Perhaps the most popular tracking is still the classic motion PIR (Passive InfraRed sensor). Those sensors are cheap and simple to use, but on the other hand they tend to occasionally misfire and often lack precision for complex applications. Still, they are part of many rule-based systems due to their simplicity.

The second type of tracking, concerns the tracking of daily life objects. Object tracking is another non-intrusive source of information that can be exploited in a smart home, which provides a much richer data set with temporal continuum. It is a major improvement from the simple set of discrete Boolean information discussed before [40] as it embeds both high-level spatial and temporal information about the tools used by the resident. While the tracking of the resident is a topic that has been vastly explored [104, 92, 41, 79, 75, 110, 36], the story unfolds very differently for the indoor tracking of the objects used in daily activities.

The widely used technologies for resident tracking (and robot tracking [51]), cannot be directly used for the tracking of daily life objects. Indeed, all these technologies require some sort of device to be installed on the objects (either a wearable, an antenna or a tag). Installing wearable on every tangible object of a smart home is obviously not economically viable, and BLE (Bluetooth low energy) beacons or active tags are too big to be installed properly on most objects. There are few object tracking systems. Existing ones commonly involve computer vision [127] which poses an invasiveness problem in private housing facilities like smart homes. The passive counterpart of RFID technology can offer an alternative solution. Passive tags, which do not hold any inner power supply, are much smaller than active tags (1mm thick) and can be installed on most objects. In addition, these tags are very inexpensive and, therefore, an affordable solution to the context of smart homes.

Still, object tracking with RFID offers specific challenges and applications when compared to classical resident/robot tracking. For example, it can sometimes be difficult to put a tag on some objects, either because of their form, or their usage, or their lifetime. While we can minimize the effect of those challenges, there are some others that are harder to solve due to the nature of the technology and the materials involved. In fact, most tags do not resist high temperature and cannot be used in the oven as inner components can melt. Microwave oven can also be problematic as it destroys the tags. Perishable objects like food can obviously not receive tags. Moreover, adding tags to several objects can increase the interference in a small room. There is also a physical limit on the number of tags a reader can process. For those reasons, object tracking is more challenging than human/robot tracking with wearable.

Despite all the challenges mentioned, object tracking is very promising for a better granularity in activity recognition in smart home and therefore it seems to be the most logical and reliable choice for the future of that field. It has been shown in previous studies that knowing the position of all objects at all time enable accurate step by step activities of daily living (ADL) recognition [20].

Since this final expert system will be used within a smart home for cognitively impaired persons, tracking objects is a more reliable way than tracking human using wearable as those persons can easily forget to wear them. This is also why we chose to use passive RFID tags. By putting them on objects, they should always be available and provide accurate data. However, it is possible to track both humans and objects at the same time using the system described in this paper.

The contributions of this paper are three-folds:

- The datasets collected for this project in a realistic smart home infrastructure are provided to the scientific community to help the advance of this discipline. Labeled RFID datasets of ADLs are very rare in the community.
- A new indoor tracking system based on the relative positions using passive RFID is introduced. This system can track several objects in real-time. Very few approaches can do this task in the literature.
- Three filters that can stabilize RFID readings and positions are introduced for this system. The filters are simple, yet very efficient according to the experiments conducted in our smart home.

This paper contributes to expert systems in the fields of healthcare and information retrieval. It uses classical data mining techniques to locate objects in a smart home and custom filters to track their positions over time. Concretely, the systems proposed in this paper have two main purposes: (1) direct tracking of objects for distracted people and (2) main building block for a non-invasive activity recognition system. As stated before, a non-invasive activity recognition system is an essential intelligent system to have in a smart home designed for impaired or semi-autonomous persons.

Some examples of use of this expert system, from a medical point of view, may include:

- Detect if the person did not take his medication by looking if the pill dispenser moved (Movement is not sufficient to assert the pill was really taken).
- Fall detection by embedding tag in clothing.
- Quality, variance and complexity of the alimentation.
- Change in day to day habits following a stressful situation.

Those are all tasks that are currently performed by a caregiver and their automation would reduce pressure on healthcare systems. The final proposed system which is implemented in a full-scale apartment uses simple algorithms where every decision can be explained to a human examiner. In a medical point of view, the ability to explain a decision is often mandatory.

The remainder of this paper is as follow. The next section presents the most recent approaches in the literature about indoor tracking and assesses their advantages/disadvantages in the specific case of object tracking for an expert system piloting an assistive smart home. Then we describe how we build the indoor positioning system (IPS) and how we evaluated it. Follows our indoor tracking system (ITS) with the same steps of methodology and evaluation, with a discussion.

7.2 Related work

In this section, we present the main approaches to passive RFID positioning and tracking. The decision to ignore the literature on ultrasonic sensors, passive infrared and camera-based tracking is purely motivated by the inadequacy of these technologies in our context as stated in the introduction. Nevertheless, this section still includes some approaches that cannot be applied to objects tracking simply for comprehension purpose.

Many algorithms found in the literature are based on the reference tags principle (or landmarks) first exploited by the LANDMARC system [114]. The basic idea, shown on Figure 7.1, is to exploit the Received Signal Strength Indication (RSSI) of nearby tags fixed at known positions to adjust the RSSI of the tracked tags. The method can be improved with various statistical filters [30]. They can also be used in combination with dead-reckoning, a method that infer the next position by combining speed and direction to the last known position. [97]. LANDMARC based systems work very well in general, with a good trade off between precision and accuracy. However, they require some type of landmarks on the floor which might not be very adapted to a smart home, which encompasses furniture and where most of the tracking is not performed at floor level [51]. Moreover, this technique can hardly be adapted for the tracking of several objects.

Figure 7.1

Reference tag method

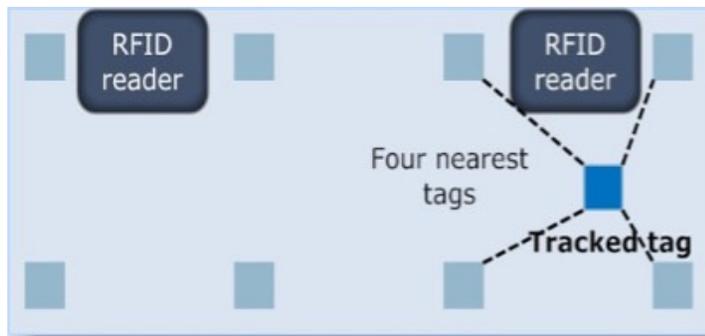


Figure from Bouchard [26]

The second family of algorithms is based on trilateration [67] and triangulation principles [101]. Trilateration uses mathematical propagation models of radio waves to convert RSSI to distance from antennas. Those distances then allow to compute isolines around antennas. The position of the tracked entity is the intersection point between three circles. Triangulation cannot be performed with all RFID systems as it requires the capability to calculate the angle of arrival. It is performed by using angles of arrival from at least three antennas in order for an intersection point to be found. The multipath propagation problem arises with those methods as the same signal can reach the reader from multiple directions with varying strengths. With this family, it is hard to track objects when they are in the range of less than three antennas. With only two of them, there is two intersections and contextual information are required to pick the right one. For instance, one intersection could be outside the room and therefore discarded. When there is only one antenna available, those systems become proximity-based as there is only one signal to use. With strategically placed antennas proximity-based methods can provide useful results and even full tracking system [90]. In addition, these techniques rarely work straightforwardly with passive RFID. They often produce several intersection points or none at all. Consequently, ad hoc method must be exploited to handle each of these special cases.

Finally, a last family exploits data mining and learning algorithms. However, the work on this family is scarce and mainly explores other wireless technologies. For example, Yim et al. [161] exploited wireless local area network access points to build a decision tree during an off-line phase in order to determine the user's location. They have shown that their technique is simpler to implement and perform better than the classical fingerprinting methods. Our hypothesis is that it should also perform well with RFID technology. One advantage of data mining algorithms over trilateration approaches is that they can track objects even when they are in the range of only one or two antennas as this is what the classifier learns for this position. Also, they generally require less calculation in the on-line phase and thus, they are faster. Still, the set-up time can often be longer.

One recent example of approach using learning methods is Calderoni et al. [36]. In their work, they developed a localization system based on active RFID to track the patients of an hospital. The patients wear a bracelet and the raw signals received by the antennas deployed through 48 rooms are directly used in a random forest classifier. Although their work is interesting, they used active RFID which is inadequate in our context and their experimental dataset is very small (14622 observations).

7.3 Static positioning inside a smart home

This paper describes a concrete implementation of an object tracking expert system within a real smart home. This system is able to track in real-time several standard objects that are used in daily life activities (cups, glasses, plates, remotes, etc.). The goal is to gather more spatial information to enable higher granularity in the activity recognition algorithms. The first component of the tracking system is a module able to perform static positioning. This module, called the Indoor Positioning System (IPS), is built to work very fast. Roughly, it takes RFID readings as inputs and it outputs a position using a classifier learned with a classic data mining algorithm, the random forest. This section first presents the smart home infrastructures used throughout this research project. Then, a special object, used to build the classifier in this experiment, is presented, followed by a formal description of the logical position concept exploited by our tracking system. The last subsections are about our methodology, from data collection to model evaluation and selection.

7.3.1 Smart home

This research took place at the DOMUS laboratory, at the Université de Sherbrooke. The DOMUS laboratory contains a realistic smart home infrastructure in which people could comfortably live. The smart home is composed of six rooms: a hall, a bedroom, a bathroom, a kitchen, a lounge and a lunch-room. In each of those rooms there are many sensors including passive infrared motion detectors, standard video cameras, temperature sensors, flow-meters, pressure plates and smart power analyzers. There are also twenty RFID antennas placed strategically to cover all the inner surface of the smart home. Figure 7.2 is a picture and Figure 7.3 is a map of the apartment. The DOMUS laboratory mainly focuses its researches on helping elders to stay at home. All those sensors are used as information input to build monitoring system to accompany the inhabitants in their daily lives. Thus, being able to track objects can enable better monitoring by adding additional information about the context of an activity.

The positions of the twenty RFID antennas are shown by the big crosses on Figure 7.3. The antennas are fixed on the wall at about one meter high. The antennas are MT-262013/TRH/A/K from Wireless Edge LTD and they operate at a frequency between 902-928 MHz. The only information they give is the signal strength (in Db) and the ID of the tag that responded. There are usually between two to six antennas in range at any position in the smart home. The twenty antennas are coupled with 5 readers that operate in round robin. It means that the signal of any antenna on the same reader cannot overlap, thus reducing the overall interference level in the smart home. Moreover, to reduce the overlap even more, the antennas are configured to work at a lower range than their full capabilities (around 3m). Placing all antennas in round robin on a single reader would have the major consequence of drastically increasing the delay between two readings and therefore reducing our real-time tracking speed. As the reader see farther in the paper, the exact location of the RFID antennas is not of critical importance. As a guideline, for the method to work best, they should be grouped by readers as much as possible and there should be more than one antenna capable to reach a tag at any position in the smart home.

Chapitre 7 – Tracking objects within a smart home

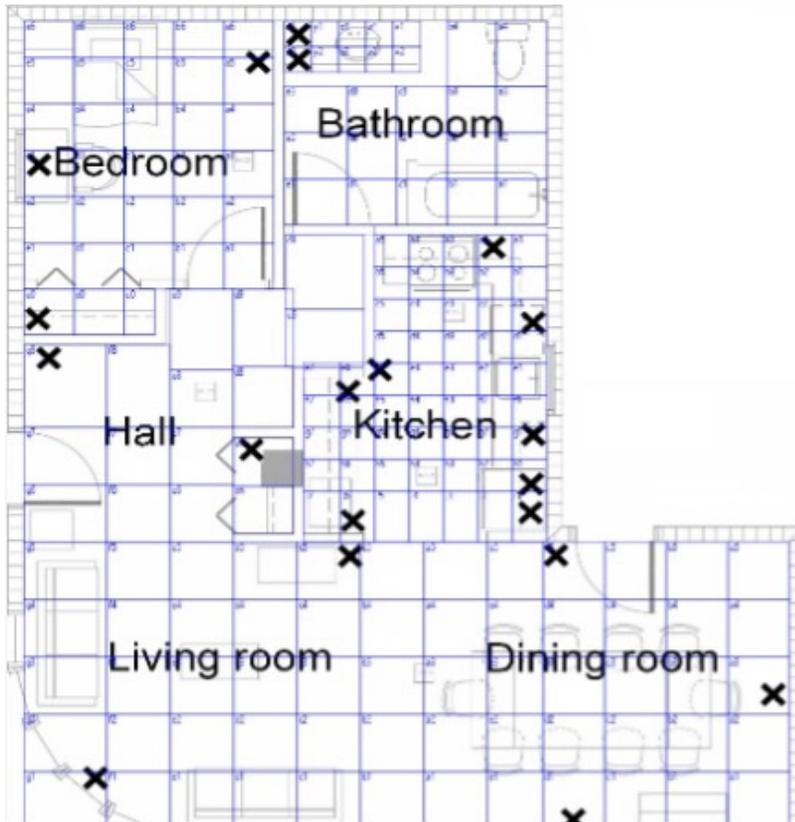
Figure 7.2

A picture of the DOMUS smart home



Figure 7.3

A map of the DOMUS smart home



The crosses mark the positions of the RFID antennas. The grid represents the zones.

Chapitre 7 – Tracking objects within a smart home

Figure 7.4

A picture of the custom object



7.3.2 Special object

As the first module toward the deployment of a real-time tracking system, the Indoor Positioning System (IPS) aims to find the position of objects within the smart home using the RSSI of RFID tags. The first step would then be to collect data in order to do the learning. To do so, it was decided to design a special object that would maximize the signals transmission. This special object has a shape similar to a plausible object findable in a house. Yet it is built to be optimal towards RFID to make sure we get the best IPS possible. The reader should note that the special object is not a requirement, and is mostly used for the learning and the testing of our tracking system. Given those constraints, an empty reusable plastic bottle of about thirty centimetres high was selected as a starting point. The bottle is in plastic so it does not interfere too much with the RFID signal. Four passive RFID tags are installed on it, each one facing a different direction. This way, a tag is almost always directly facing an antenna, resulting in the best signal possible under realistic conditions. Before putting the tags on the bottle, each of them was tested separately to make sure they all give similar signal strength. This is a required step to ensure that we discard bad tags that deviate too much from the mean, as suggested by Brusey [19]. Programmatically, we merge the RSSIs of those tags into a single one by keeping only the highest values of the four. Figure 7.4 is a picture of the bottle.

7.3.3 Logical zones

The positioning task is generally considered as a regression problem. Therefore, most systems aim to produce precise coordinates from a given origin as a position. For instance, it is the case for the GPS system, which outputs precise latitude and longitude on Earth's surface. Still, it is not always mandatory to have the exact location of an entity. Often, it is enough to have a relative position, especially for qualitative applications. Indeed, if your goal is to tell that 2 objects are within a given range, relative positions are as good as exact positions.

Moreover, in the case of an inaccurate technology, a stable qualitative information about location is more reliable and easier to take advantage off than a constantly moving precise location. With that thought in mind, in this paper, the task of positioning is considered as a classification problem where the goal is to classify RFID readings into qualitative zones. The zones are fictional divisions that can take any shapes or dimensions. Such zones are learned through a machine learning technique. For simplicity, square zones of equal dimensions within a given room were mainly used in this project. In a previous version of the IPS, heterogeneously sized zones were tested to reflect the lower quality of the radio signals in some part of the smart home, but the research team could not confirm that such measure would significantly increase the accuracy of the IPS. There are no overlapping zones between rooms even though it would not make any difference, as rooms are not taken into account explicitly in the IPS.

For the logical zones, many dimensions were tried in various rooms to see how the accuracy of the classifiers would reflect on our choice. For instance, in the dining room, the zones were first set at 100 cm x 100 cm. Then, we reduced the dimensions to 75 cm x 75 cm, 60 cm x 60 cm, 45 cm x 45 cm, 30 cm x 30 cm and 20 cm x 20 cm. At this point, accuracy dropped by a large margin and we decided to stop. This gave us an idea of the precision the final system could achieve. Nevertheless, there seem to be a general consensus around smart home researchers that an accurate system is preferable over a precise system that fails more often. The students and researchers at the DOMUS laboratory expressed the same concern in that matter. Accordingly, bigger zones are selected for the tracking system. Thereby, the following dimensions are used throughout the smart home: 40 cm x 40 cm in the kitchen, 60 cm x 60 cm in the bathroom and the bedroom and 75 cm x 75 cm in the hall, the lounge and the lunchroom. There is an exception on the bathroom counter where zones measure 30 cm x 30 cm. Those dimensions reflect the kind of activities people usually do in those rooms in term of movement amplitude.

Indeed, cooking requires smaller movement of objects than vacuuming for example, thus smaller zones are required to see those objects move. This is why the kitchen and the bathroom counter have a higher precision. Figure 7.3 presents a map of the smart home with the final zones drawn on it. In the dataset we provide, zones are named by a letter and a number (like a1). Their name could instead reflect their position, like sink1, to facilitate usage of an expert system based on logic rules on the computed positions.

7.3.4 Data collection

As we mentioned at the beginning of this section, the IPS uses a random forest to classify RFID readings to zones representing objects position. To construct the random forest classifier, a learning dataset was constructed. Since there are twenty RFID antennas throughout the smart home, a reading consists of a vector of twenty integers, one per antennas. The possible values range from -70 decibel to -35 decibel. For each zone, we collected fifty readings by placing the custom bottle in the middle of them, at about one meter height. This height is the one at which most people hold objects while moving them. The bottle is in the middle of each zone to get the most distinct readings possible. In no way it means that to position objects they have to be in the middle of a zone, it was simply done to maximize the readings difference from zone to zone during the learning phase. The special object is placed on a wooden stool to avoid interference from metallic surfaces or human body. Also, the antennas were set to emit a signal with a 750ms interval to avoid interference between them. Using the final dimensions, this means a total of 9550 readings in 191 zones. They are all available on the DOMUS website¹, along with the 25 000 more readings we collected during the precision test described in the previous section in order for researchers to reproduce our results or work toward improving the indoor positioning of objects with passive RFID.

¹ <https://www.usherbrooke.ca/domus/>

7.4 Building models for the IPS

The next step to build the Indoor Positioning System (IPS) is the learning phase itself. The positioning is considered here as a classification problem. To do so, thorough testing with the most popular classification algorithms were performed. Our goal here was not to make a strong theoretical contribution, so building a specific classification algorithm was ruled out. Moreover, it turned out, as expected, that most models perform quite well on this type of classification problem. Even though the bulk of our tests was done on decision trees, other families of algorithms, such as the multilayer perceptron and the Bayesian network, were also explored. The well-known Weka data mining library [74] was exploited as well as some of its extensions in order to achieve this learning phase. The classification problem consists of finding to what class a vector of twenty values given by the twenty RFID reader belongs. The classes are the different zones, expressed in the generic format *room_number*, like *lounge_a1*. This generic format then allows to convert zone names into Cartesian coordinates if needed. Our problem is, therefore, a low dimension combined to a high number of classes. There are actually a countable finite number of possible different vector of value in this specific problem. We argue, however, that an exact model cannot be learned since two identical vector instances could belong to different classes given the precision of our RFID readers. Moreover, obtaining all different instance of vector could take a significant effort, if even possible.

Four metrics were used to evaluate the performance of each classifier. The first one, *Error dist*, is the mean distance of the wrongly classified readings from the target zone only. The second one, *Mean dist* is the global error margin on the dataset. Distances are given in zones and not in centimetres as zones do not need to have all the same dimensions. The third one, *Wrong zone* is the total number of misclassified zones. It is in direct relation with the last one as we can express the *Accuracy (Acc)* as $1 - \text{WrongZone}/\text{NbTotal}$. The remainder of the section describe the algorithms tested to build the IPS and present a comparison of their performance.

Table 7.2

Accuracy and distances with non tree models

Non Tree	Error dist	Mean dist	Wrong zone	Acc (%)
NBayes	3.1058	0.4885	1480	83.2738
NNET	2.7501	0.2128	728	90.5337
1NN	2.5408	0.1268	473	94.7573
BayesNet	2.0543	0.0684	316	96.6064

7.4.1 Multilayer perceptron

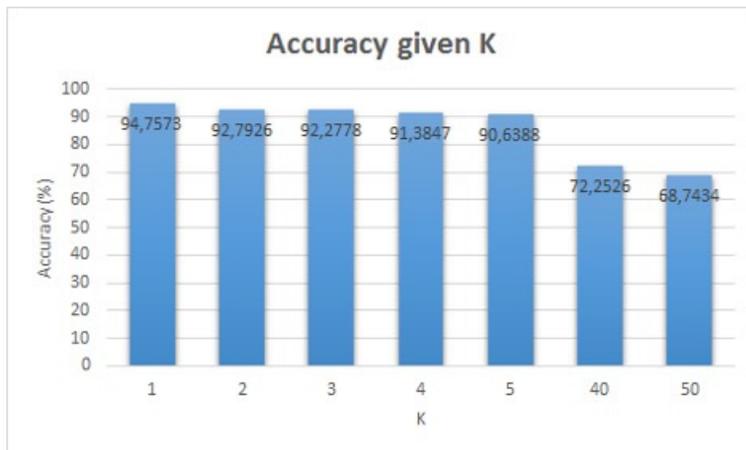
The multilayer perceptron [9], also known as a neural network, is a powerful model that is state of the art for many machine learning tasks. In Weka's implementation, all nodes are sigmoid units and learning is done through back-propagation. The loss function is a simple Boolean. It has a single hidden layer of size given by: $(NbAttributes + NbClasses)/2$. The learning rate is 0.3 and the momentum is 0.2. The network is trained for 500 epochs, unless a validation set is provided. If so, early stopping is done after 20 epochs with decreasing performance. Because of all those hyper-parameters, training a neural network can often be a long and complex process. Neural networks are good at generalizing, especially from high dimensional datasets, but this is far from being the case in our context. Finding the best hyper-parameter is also a tedious process that can take a long time depending on the computing power available. The results are shown in Table 7.2 under the label NNET.

7.4.2 *K-nearest-neighbours*

The k-nearest neighbours [6], called Ibk in Weka is a simple algorithm that consists, for each instance to classify, to look for the k-closest ones in the training set. The Euclidean distance is used to measure the similarity. The default value for K is 1. We also tried with K varying from 1 to 5. Figure 7.5 shows the results of those tests. As we can see, accuracy decreases as K increases. The two last tests in Figure 7.1 demonstrate that not all fifty readings from the same zone are the same and that some other zones have very similar readings. If all readings were the same for a zone and never found in any other zone, results would have been similar to the first ones. However, it requires the entire dataset to be contained in memory, which might cause

Figure 7.5

Accuracy given K in k-nearest neighbours problems on some architecture with limited memory



7.4.3 Bayesian network

The Bayesian network [25] is a probabilistic model presenting itself as a directed acyclic graph that we can use to represent a probability distribution of classes over attributes. Training a Bayesian network can be seen as two separate steps: learning the network structure and learning the probability tables. The one present in Weka offers numerous possibilities in the choice of algorithm for each of those steps. We used the default K2 algorithm to learn the network structure. K2 is a hill climbing method that uses a fixed ordering of variables to maximize quality measure of the network structure. In our case, the quality measure was the Bayesian metric from Cooper et al. [37] (see equation (1)), a measure that tends to approximate the likelihood of the graph. The graph was initialized as a Naive Bayes Network. This means that the classifier node is connected to all other nodes. In equation 1, B_S represents the network structure of the database D . $P(B_S)$ is then the prior network structure and r_i is the cardinality of the data. N_{ijk} is the number of cases in D where the variable x_i as the value v_{ik} .

$$Q_{K2}(B_S, D) = P(B_S) \prod_{i=0}^n \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(r_i - 1 + N_{ij})!} \prod_{k=1}^{r_i} N_{ijk}!$$

To learn the probability tables, the SimpleEstimator is used. It estimates the conditional probabilities directly using the given data. A smoothing constant of 0.5 is used by default when computing the probability tables. This model proved to be one of the best tested in our experiments, only matched by the Random Forest. However, it requires a very long training. Indeed, the complexity of complete inference is NP [155]. Classification could also be longer as well because it requires a lot more computation than decision trees which are usually under $O(\log n)$.

7.4.4 Naïve Bayes

The Naive Bayes (NBayes) model does a simple data analysis to later perform classification by a probability calculus over all the attributes. If C is the class and x a data vector from the set X , then the probability of a class c given the example x is $P(C = c | X = x) = \prod p(X_i = x_i | C = c)$. If needed, the data is discretized.

This model makes the assumption that the attributes are independent from the class. Once the probability for all classes is computed, the max is selected as the class. This model did not perform well for our IPS.

7.4.5 Classic trees

The next category of algorithms tested are the decision trees. Our hypothesis was that these algorithms would be the most adapted to the properties of our dataset. Therefore, the most popular algorithms were all tested. The results are summarized in Table 7.3. The same metrics were used for the tree-based models than for the previously tested learning methods.

First, Weka offers two implementations of the decision tree, one using the Cart algorithm [21] and one using the C4.5 algorithm [128]. C4.5 is called J48 in Weka. It uses the information gain metric to find the best attribute to split on in each node until there is only one class left or a minimum number of examples is reached. Then a pruning step is done trying to remove leaves that do not bring any accuracy gain. There is also a slightly different version of C4.5, called J48Grafted [156]. It essentially is J48 with a post training algorithm that adds nodes to an existing tree to reduce training error. However, this grafting step was not very effective on our dataset, yielding in an increase in accuracy of less than 0.1 percent. Cart is called SimpleCart in Weka. It uses the Gini impurity as the splitting criterion. It can be expressed as in equation (2) where i and k are classes from the item set in the node on which I_G is computed and f represents the fraction of the item in that set that have the indexed class.

$$I_G(f) = (ff_k) \quad (7.2)$$

$i \neq k$

Table 7.3

Accuracy and distances with the trees

Tree	Error dist	Mean dist	Wrong zone	Acc(%)
LADTree	2.5308	0.2683	1001	88.5690
Reptree	2.6336	0.2571	916	89.0523
FT	2.6242	0.1870	674	91.2797
LMT	3.7239	0.2711	673	91.5213
RTree	2.9782	0.2264	718	92.0782
Cart	2.3776	0.1645	652	92.4564
J48	2.2521	0.1376	577	93.4860
J48graft	2.2254	0.1322	561	93.5701
NBTree	2.7027	0.1486	520	93.9588
RForest	2.4331	0.0622	243	97.2158

It also differs from C4.5 at the pruning step where it uses the cost complexity pruning. It performs worst by 1 percent than C4.5, which suggests that pruning does not yield significant differences with our dataset.

Another tree that we tried is the Random Tree (RTree) [28]. In it, only a subset of attributes is considered for each split. It is therefore well adapted to datasets consisting of a large number of attributes. The size of this subset is $\log_2(D) + 1$, where D is the number of attributes. There is no pruning. It performed slightly worse than the other trees. Indeed, our datasets are composed of only 20 numerical attributes and Random Tree is better suited for very high dimensionality problems.

7.4.6 Trees mixed with other models

The trees algorithm presented in section 7.4.5 all worked under the same basic principles. This sub-section presents three other trees that have a model on their leaves or nodes.

Functional Tree

Functional Tree [68], FT in Table 7.3, is like a normal tree with the difference that it has logistic regression functions at inner nodes or leaves or both, as in our experiment. Weka uses fifteen iterations of LogitBoost [63], a variant of AdaBoost where logistic regression techniques are applied. If AdaBoost is a generalized additive model, LogitBoost is a convex optimization of its logistic loss. The Functional Tree was not very effective on our data, with a lost in accuracy over C4.5.

NBTree

NBTree [95] particularity is that its leaves are composed of naive Bayes classifiers. The author of this model affirms that it can outperform both the normal tree and the naive Bayes classifier, especially on large datasets. With a gain of 0.5 percent over C4.5 and of 10 percent over the naive Bayes, this affirmation seemed to be confirmed.

Logistic Model Tree

The Logistic Model Tree [99], LMT in Table 7.3, looks like the FT in the way that there is logistic regression functions at the leaves. It produces bigger trees than FT, thus being longer to train for similar accuracy. Indeed, a bigger tree implies more leaves and more logistic functions to learn. It also means that prediction time is slower since there are more nodes to examine.

Alternating tree

The last single tree we tried was the LADTree [80], an alternating tree capable of multi class classification. Like the Functional Tree, LADTree uses LogitBoost as the base of the algorithm. The tree presents itself as a series of AND/OR rules, one rule per boosting iteration. A score vector is associated to each answer to those rules (true or false), one score for each possible class. Classification is made by adding all confronted scores at each node reached. Then the class associated with the highest score is chosen.

By default, Weka uses only ten boosting iterations, but it resulted in an accuracy of only 31.8554% with our dataset. In their original paper, the authors show that accuracy gets significantly better with more iterations. Table 7.3 shows the results with 50 boosting iteration. With 50 iterations it gives an accuracy of 88,5690%, a number more consistent with the results from other trees.

The downside of this algorithm is that adding more iterations also greatly increases the training and the prediction time. While it took only minutes to train and test with the 10-fold cross-validation and ten boosting iterations, it took several hours with fifty.

7.4.7 Trees forest

The last tree algorithm that was tested is the Random Forest (RForest) [28]. Random Forest is simply 100 Random Trees trained separately, exactly the same way as described before. The predicted class is then the modal class between all 100 predicted classes. This forest surpassed the single random tree, thus being by far the most accurate model on our datasets. It even outperformed the Bayesian Network while being shorter to train and faster to use. Its mean distance is more than half the nearest tree which means even when a tag is positioned wrongly it has more chance to be close enough for the activity recognition algorithm. Nevertheless, it is slower than basic classifiers such as C4.5 and SimpleCART, which was to be expected since it is like searching in a C4.5 tree 100 times in a row.

7.4.8 Discussion

In the previous sections, the algorithms tested to build the IPS were presented. Most of them produce a similarly good accuracy and a low mean distance from target. Still, some of them are considerably faster to train and to use. Training time is not really a big factor since it is part of the off-line configuration phase. Prediction time, however, is important to consider for the final goal of real-time activity recognition. This final system will have to locate many objects multiple times per second for the expert system to take quick decision and act upon it. For this reason, we chose the use the random forest as the classifier.

Classification in a random tree is really fast as it requires few numerical comparisons. The maximal depth of a tree is 17 nodes in our final random forest for a size of about 2900 nodes and leaves. The model built can be used straightforwardly to classify any objects equipped with passive RFID tags into one of the logical zones. The RSSI values of that object simply have to be passed as a parameter to the model into a single 20 attributes vector. If the object is equipped with more than one tags, the signals are merged using the method described in the previous section. For the final IPS, we chose to use a random forest of 50 trees (97.2999% accuracy). By using only 50 trees, the classifier is faster for an accuracy similar to the 100 trees. Both training and classification are faster, which is the principal criterion for this part, given similar accuracy.

7.5 Dynamic tracking system

The next part of this paper focuses on transforming the accurate indoor positioning system into a dynamic system able to track several objects in real-time. It first presents the software architecture then the experiment we did with it.

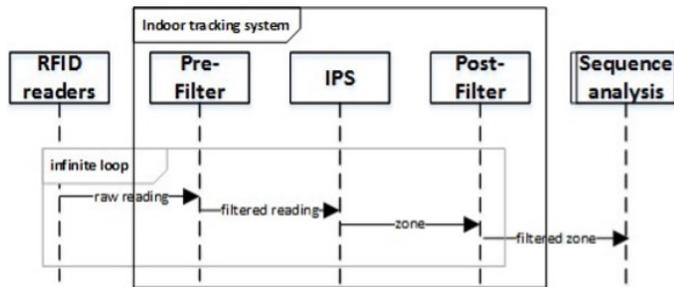
The indoor tracking system (ITS) is built around the IPS. It handles all readings from the collection to the final zone and gives it to the sequence analysis module. The complete flow is shown in Figure 7.6. First, there is a module responsible to acquire readings from the readers and aggregate them by object, if necessary, as explained previously. From this point, a data object consists of the reading vector, the object's name and a timestamp. The object's name is retrieved from a database using the tags ids. The ITS uses this name to keep track of objects over time as each inner module works asynchronously.

At the beginning, the ITS only served to record the objects' positions over time. However, first results quickly showed the need for some filter. For this reason, we added a first filter (pre-filter) on the raw readings to reduce the impact of a misreading. Then, a second filter (post-filter) is applied to the zone found by the IPS the ensure regularity over time. As those filters need context to operate, they are included in the ITS. This way, they have access to the full sequence of positions, or the required window.

The ITS is currently implemented in Java. Each module works in his own thread and they exchange data using a custom message queue with the publish/subscribe paradigm. Execution speed is an important factor to maintain the real-time aspect of the tracking. Accordingly, each module is optimized to do its task under 20 milliseconds, which is the speed of the RFID readers. In fact, on average, they operate for about 4 milliseconds per object and sleep until a new reading awakes them.

Figure 7.6

Data flow inside the ITS



The sequence analysis module is not properly part of the tracking system. Its only purpose is to extract statistics on the system. Since the ITS is always running, it is in the sequence analysis module that we can record the paths taken by the objects as determined by the ITS and compare them to a theoretical path.

In Figure 7.6, there is no mention of any user. This is because the whole system is always working, always analyzing the position of every tag. If an object starts moving, it should be because someone moved it, but it still does not make this person a user, only a source a variation in the system, much like the sun is not a user from the perspective of a thermometer.

7.5.1 Filters

As stated before, the first results indicated a need to filter the data received from the RFID readers to make the predicted zones and sequence more compliant with the reality. Indeed, even if the accuracy of the random forest is around 97% at 750 milliseconds interval, there are still some interferences that occur occasionally. When the antennas emit at 20 milliseconds, those interference happen more often, leading to more positioning errors.

Filter on RFID readings

The first filter, noted pre-filter in Figure 7.6, exists to regulate the raw data. It is needed because sometimes a reader might give a value of 0 Db when it should not because of interference. This wrong value could then conduct the classifier to predict a wrong zone. This kind of error made objects move by themselves when they were not supposed to be moving at all. Instead of just ignoring the whole reading, we choose to compensate for the missing values by using a moving average on each reader. It is expressed by the following general expression:

$$\bar{R} = \frac{\sum_{n=0}^{N-1} C_n * R_{N-n}}{\sum_{n=0}^{N-1} C_n}$$

Where R is the weighted average, R_{N-n} is the N th reading and C_n is the C th weight coefficient. Note that we consider N_0 as being the current reading. We also have to divide by the sum of all coefficients to make sure they sum to one. We tried four different weighting function. The weights are given by the following functions: the logarithm function, the linear function, the exponential function with base 2, and finally, the factorial function. We also tried the simple moving average, or the constant function. They are explained in Table 7.4. The algorithm for the filter is simple :

1. Receive a vector from the previous module;
2. Add the vector to a circular buffer of size n ;
3. Apply equation 7.3;
4. Send the result to the next module (the IPS).

Filters on predicted zones

The second filter, noted post-filter, applies directly on the predicted zones. While the first filter can reduce the impact of some misreading, there can be long interference in the signal that extends the moving average window. There is also some situations where the objects seem to teleport themselves, also because of interference. This second filter directly tackle this issue. It comes in to version: a blocking filter that only allow a predicted zone to be a direct neighbour from the previous one and a limiting filter that only allow moving one zone at the time.

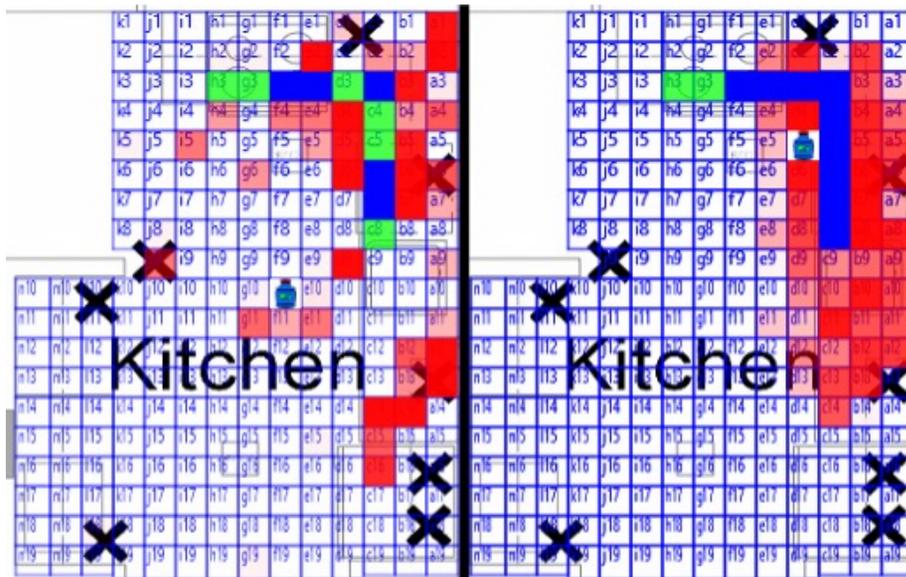
Table 7.4

Weighting function for the pre-filter

Distribution	Expression	Remarks
Logarithm	$C_n = \ln(1n+2)$	We use $n + 2$ to avoid the part of the logarithm function that is negative or rapidly growing.
Linear	$C_n = 1 - 0.1n$	
Exponential	$C_n = \frac{1}{a^n}$	
Factorial	$C_n = \frac{1}{a \cdot n!}$ I: $\frac{1}{n}$	We only use $N = 10$ and tweak the modifier a as $n!$ grows too fast.
Constant	$R = \frac{N - n}{N}$	

Figure 7.7

Map of the kitchen with raw data versus the limiting filter



The green squares show the expected path and they turn blue when correctly identified by the algorithm. Red squares are wrongly identified zones.

Figure 7.8

Example of the filters



In green is the current zone and the one predicted by the blocking filter. In red is the zone predicted by the limiting filter while in blue is the zone predicted by the random forest.

The blocking filter is the first one we tried with this approach. Let us say the object is thought to be in zone A2 and the next predicted zone is D5. Of course, one of the two positions is false as this would mean the object teleported, assuming it moved at normal speed. In that case, the blocking filter would reject D5 and say the object is still in A2. Illustration of this example is shown in Figure 7.8. Obviously, this means that if the filter misses the first move for some reason, the object would get stuck. To resolve this issue, we added a time-out on the position where after X rejected zones the current position becomes the predicted one, no matter what the filter says. In our experiment, we used a $X = 20$. This represents about half a second which is the time needed to cross most zones at average speed. The algorithm for this filter is :

1. Receive a zone from the previous module;
2. Retrieve the previous zone;
3. If the received zone neighbours the previous zone, return the zone;
4. Else if the time-out is expired, return the zone and reset the time-out;
5. Else return the previous zone and increase the time-out by one.

The limiting filter is a more permissive version of the first one where moving is always allowed, but one zone at the time. This means that with the previous situation where the object is in A2 and the newly predicted zone is D5 the filter would output the zone B3, as illustrated in Figure 7.8. This filter has eight degrees of liberty that allow moving in each direction of our two dimensions grid. Since moving is always allowed, there is no need for a time-out. This means that every outlier will produce a move. In other words, this filter does not remove any outlier, it only reduces their impact. On the other hand, it increases the speed at which the system can recover after missing several zones when compared to the blocking filter.

Moreover, the fake zone visitation it creates have a high probability of not being fake at all since objects tend to move in straight ways, either when rolling, falling or when held by humans. Figure 7.7 shows this filter in action. The algorithm is :

1. Receive a zone from the previous module;
2. Retrieve the previous zone;
3. Compute a combined zone:
 - a. Move the x axis by one unit in the direction of the received zone;
 - b. Move the y axis by one unit in the direction of the received zone.
4. Return the combined zone.

7.6 Experiments

In this section, we present the experiments we conducted using the ITS described above. The results were all collected by the sequence analysis module.

7.6.1 Tracking data

As with the IPS, the first step for the indoor tracking system, ITS, was to collect meaningful data we can compute the accuracy on. Since the final use of the ITS will be as a key component of an activity recognition expert system, meaningful data is when you mimic a real-life activity. So, for each room, we designed four paths representing activities like taking something from the buffet to the kitchen counter or entering the apartment, going to the closet before exiting. We draw those 24 paths on a map of the zones and listed crossed zones. This forms the theoretical dataset. The practical dataset was collected by making the same plastic bottle of water we used for the IPS follow the various paths. To do so, we placed black tape on the floor and we walked the paths holding the bottle in our hands. At all time, we made sure the bottle was right over the line. To add some realism, we varied the position of the bottle by sometime holding it upside down or on the side. We repeated each path ten times to get more data. For this experiment, RFID antennas were set to emit at 20ms intervals to better catch the movement. However, RFID is not a real time technology. This means the system will do its best to emit at 20ms interval but it may take longer. There is no guarantee on the real time between two RFID signals. The practical dataset then consists of several thousand RFID readings for each of the 24 paths. Figure 7.9 shows our experimental set-up, with the tape on the floor and some of us holding the bottle. This dataset is also freely available on the

Figure 7.9

Experimental set-up



DOMUS website. It is important to understand that these paths were used in order to accurately evaluate the system. The ITS is in no way limited to tracking these examples.

7.6.2 Evaluation metrics

Once we had collected the dataset, we wanted to evaluate how the ITS would perform on it. Literature on how to evaluate a tracking system is scarce. Most of the time, it is done manually, or by comparing to a highly accurate tracking system such as ActiveBat [76]. A system like that is expensive to buy, and in our case, since we are tracking objects and not human, there is no guaranty it would have worked properly. In the end, we used two metrics called Targeted Zones Found (TZF) and Sequential Targeted Zones Found (STZF) to evaluate it.

Let us say that T is the set of all theoretical zones for a given path and that P is the set of the practical zones found for this path. Then, TZF corresponds to $T \cap P$, or, to put in simple words, TZF is the proportion of the theoretical zones the classifier correctly identified. Logically, if the classifier was perfect this score should always be perfect since this is essentially the metric it was trained on.

The STZF metric is an extension on the previous one that takes into account the temporal factor of a tracking sequence. For this metric, a predicted zone will only be counted if the previous theoretical zone was previously identified, as in a finite state machine. For example, imagine there is a theoretical sequence formed by the zone B-A-C, in that strict order. Imagine also a practical sequence of A-A-A-C-A. In that case, the TZF would be $2/3$ and the STZF $0/3$ since B was never predicted. If the practical sequence is A-A-A-C-B-A, then TZF is $3/3$ and STZF $2/3$.

Another metric commonly used while comparing sequence is the edit distance, also known as the Levenshtein distance, which reflect the number of operations needed to convert one sequence into another. This metric is not really significant in that case since the practical sequence is much longer than the theoretical one giving a poor score due to the number of *delete* operations to perform.

7.6.3 Results

The previous section presented the two metrics we use in this section to characterize the ITS. We first present the raw results and discuss on them. Then, we explain the filters we added in order to increase the accuracy of the system.

Raw results

In the beginning of this section, we presented the tracking dataset we collected, then the metric to evaluate it. Figure 7.10 and Figure 7.11 show the raw results of the ITS, given per room. Given the high accuracy of the IPS, those results are poor and clearly indicates that the human body greatly interferes with the RFID signal. With an average of 36% accuracy on the STZF metric, the system could not provide much useful information about objects' ongoing movement, only the position before and after.

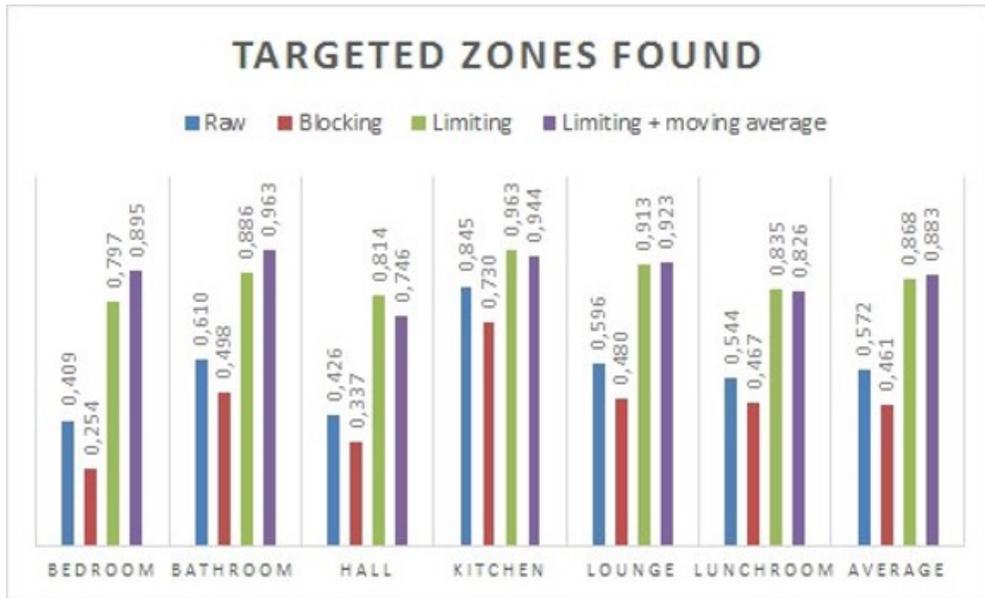
Results with filters

The first filter we added to the ITS is the moving average discussed before. Results for all weighting function are shown in Table 7.5.

It clearly shows that the accuracy increases with N in the logarithm coefficient, while still staying under the accuracy of the raw data and therefore, also under the accuracy of the linear moving average. For the exponential coefficient, we tried many values for the modifier a . With $a = 1$ the results are terrible. However, with $a = 1/8$ or $a = 1/16$ the results are much better. The constant coefficient is interesting in that it gets its best results with a low N , as opposed to all other function. This suggest that at our walking speed the filters should give less importance to older readings.

Figure 7.10

Results for the targeted zones found metric



Here are some results with some parameters on the kitchen paths. Note that accuracy increase with N , and the more smooth the function the better. Result for the post-filter are shown in Figure 7.10 and in Figure 7.11.

Combining filters

So far, we presented two families of filters we can use to improve the accuracy over raw RFID readings. The first family alters the raw readings to smooth them with a moving average while the other compares a newly predicted zone with the current one to compute what should be the right zone.

As they work on different sides of the same problem, they can be combined without any interference between them. We tried all possible combination and some results are shown in Table 7.6. Results show that the best combination is the limiting filter with the moving average

Figure 7.11

Results for the sequential targeted zones found metric

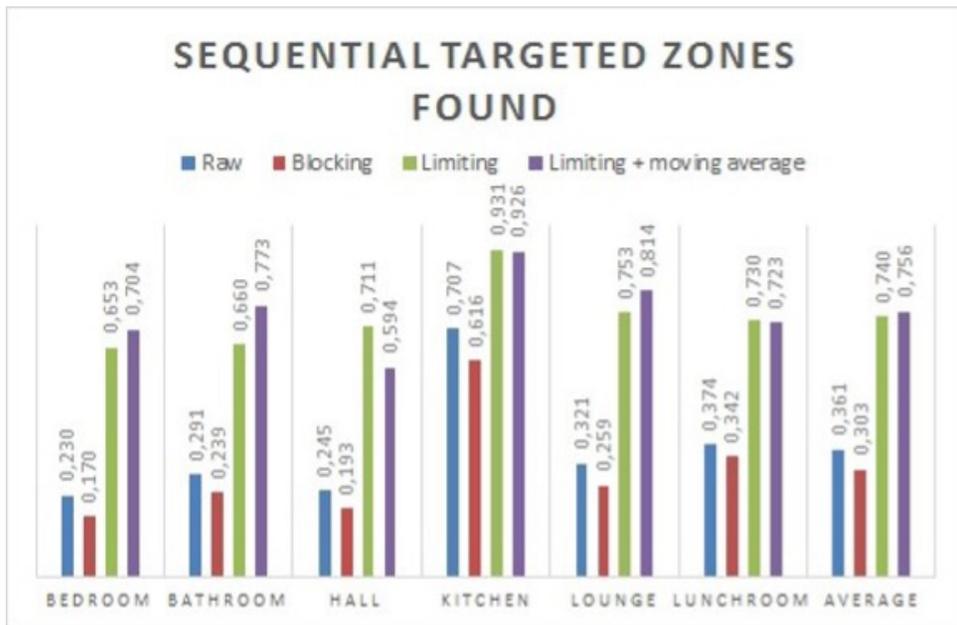


Table 7.5

Effect of N and the weight function with the moving average in the kitchen

Parameters		Metric	
N	Weight function	TZF	STZF
5	Constant	0.814	0.656
1	Constant	0.772	0.652
0			
2	Constant	0.745	0.575
0			
3	Constant	0.687	0.518
0			
4	Constant	0.673	0.496
0			
5	Log	0.367	0.288
1	Log	0.489	0.374
0			
2	Log	0.593	0.430
0			
3	Log	0.565	0.385
0			
4	Log	0.552	0.400
0			
5	Linear	0.417	0.315
1	Linear	0.533	0.393
0			
2	Linear	0.763	0.648
0			
3	Linear	0.783	0.600
0			
4	Linear	0.789	0.565
0			
5	Exp a=1/8	0.426	0.329
1	Exp a=1/8	0.666	0.498
0			
2	Exp a=1/8	0.685	0.493
0			
1	Exp a=1/16	0.661	0.499
0			
2	Exp a=1/16	0.733	0.580
0			
1	Fact a=1/4	0.623	0.455
0			
1	Fact a=1/8	0.821	0.677
0			
1	Fact a=1/16	0.886	0.776
0			
1	Fact a=1/32	0.875	0.756
0			
1	Fact a=1/128	0.878	0.720
0			

Table 7.6

Limiting filter with the moving average

Metric	TZF	STZ F
Bedroom	0.820	0.650
Bathroom	0.946	0.827
Hall	0.762	0.514
Kitchen	0.970	0.937
Living room	0.926	0.754
Dining room	0.843	0.734
Average	0.878	0.736

using the factorial distribution, as shown in Figure 7.10 and in Figure 7.11. They were both the best filters of their family and they work well together, bringing an overall increase of 27% over the raw accuracy.

7.6.4 Results

In the previous sections, we mentioned several times that our system can do real-time localization. Given the nature of the algorithms in use, it is not possible to do hard real-time, that is when we can count the exact number of operations performed. However, it is possible to put a limit on the required time for each module. In our implementation, this limit is still relative as we use classic Java and let the garbage collector runs when it seems fit. Empirically, we measured some statistical indicators on our datasets by running them in the ITS. Results are presented in Table 7.7. The longest the ITS ever take on our data is about 13ms. Considering that data collection happens every 20ms in an asynchronous fashion, even the longest data is still classified fast enough, with a 7ms in reserve. In average, the ITS takes about half the available time.

Chapitre 7 – Tracking objects within a smart home

The pre-filter and post-filter modules are faster than the IPS, at about 2ms for the best configuration presented above. As they also execute in their own thread, they use 10% of their budget time. The IPS is the slowest module and therefore is the one that might need optimization before being used to track every objects of a smart home. A C++ implementation instead of a Java implementation could help if the need arises.

Table 7.7

Execution time of the ITS

Room	Max time	Min time	Mean time
Bedroom	12.3207ms	0.0429ms	0.0753ms
Bathroom	6.6098ms	0.0429ms	0.0734ms
Hall	7.8145ms	0.0308ms	0.0465ms
Kitchen	11.1712ms	0.0604ms	0.0945ms
Living room	12.9440ms	0.0489ms	0.0839ms
Dining room	13.1949ms	0.0305ms	0.0513ms

7.7 Discussion

In the previous section, we presented the results of our algorithm on a moving object. Results show that the accuracy of the system drops by a large margin. However, filters can be used to diminish the impact of interference. We applied a first filter directly on the RFID readings to try to smooth variations between them. Physics tell us that the power of a wave diminishes as it moves away from its source. This variation is always monotone. Accordingly, no RFID reading should ever change drastically if the tag moves normally. Knowing this, it is normal to consider bad readings from interference as outliers and to try to smooth them using a moving average. In a future work, it could be interesting to try to infer correct readings to replace those outliers, like with a Hidden Markov Model. The results also show a greater accuracy in the kitchen. This room has the highest concentration of RFID readers of the smart home. Accordingly, this is the room where the signal is the strongest and where the interference is at its lowest.

From the experiments we conducted, it appears that the choice of the classifier does not have a big impact on the positioning accuracy. Many classifiers present an accuracy over 90%. Trees, especially, all produce similar results. This implies that rule-based methods in general can solve the indoor positioning problem when presented as a classification problem. This is an advantage over probabilistic methods as they offer a better traceability of the decision. Another implication based on our experimental outcomes is also the main drawback of a tree classifier: what happens if an antenna gets a bad reading? A bad reading, no matter the cause, can prevent a rule to be met and may then result in apparent teleportation. We could mitigate this problem by building a tree where the physical proximity of the leaves is taken into account.

Our systems offer many strengths. They are summarized in the five following points :

- real-time localization thanks to the low complexity of the classifier;
- high accuracy which makes it possible to find weak spots and act accordingly (add an antenna, move metallic structure to reduce interference, etc.);
- multiple objects concurrent localization. The limit will come from interference at the readers, not our algorithms;
- scales to many more zones and many more antennas. The given model can take much more than 20 antennas, especially if their range have minimal overlap;
- arbitrary chosen precision. The IPS can mix precision of many meters in a warehouse and few centimetres in the adjacent office in the same system, for instance.

7.7.1 Effect of speed

We said earlier that our IPS was trained with the antennas set at 750ms between readings and that we reduced it to 20ms for this experiment. Such a change could have greatly affected the accuracy of our IPS. To make sure it did not, we conducted a simple experiment. We designed a special path over the kitchen counter and we walked it with the water bottle two times, one at normal speed with antennas set at 20ms and one at very slow speed with antennas set at 750ms. Our simple experiment indicated that the accuracy was better at 20ms when no filters were used. However, this is certainly because we were still moving too fast when at 750ms between readings, reducing the chance for our random forest to predict the right zone at least one time. Still, the difference in accuracy was small. Further experiments should be conducted on this subject.

7.8 Conclusion

In this paper, we presented an indoor tracking system for objects. This ITS addresses the problem as a classification problem and uses a random forest to predict relative positions. The random forest has an average accuracy of 97% on static objects. With the help of some filter, the whole system has a tracking accuracy of 75% on the sequential targeted zones found metric with moving objects. One big advantage of this system is that it does not require any calibration and it uses fast algorithms, making it suitable for real-time tracking. It is based on a low cost and non-intrusive technology, passive RFID. It is designed to enable fine-grained activity recognition. Despite those strengths, this ITS still has some weaknesses. The two biggest are the training phase of the ITS and the effect of human interference on the readings. Effectively, it takes significant time to collect the data needed to train the random forest. Future work on this subject could be to develop an automated solution, like a robot, to record all needed readings. One way to overcome the human interference problem could be to increase the strength of the signal, which was not possible with our antennas. It would also help to add more antennas in a way that the object is always directly facing an antenna despite being held by a human. Some new filters might also help on this subject.

The research presented in this paper opens several new research directions. A first one would be to find a way to reduce the offline learning time. A way to do this could be to look how many training examples are really needed. A second direction might be to introduce a reference tag at a fixed position (in a cupboard door, for instance) to propose a hybrid method between our system and the LANDMARK system. As indicated before, the filters we designed are very simple. More filters could be created to increase accuracy. There are many general filters for time series that we have not tried, like the Kalman filter. In this work, we only used the raw readings from the RFID antennas. Future work could explore advanced features extraction from the datasets we provide.

Feature considering the readings as a time series might also improve the tracking by naturally smoothing bad readings. As mentioned before, future research on trees could try to regroup leaves based of the physical proximity of the class they represent. This way, we think apparent teleportation could be avoided as a bad reading at one antenna would likely produce a classification closer to the right zone. Finally, the future work we intend to do is to use this tracking system for activity recognition within the DOMUS laboratory. The resulting expert system will most likely be coupled with a cooking assistant for people suffering from a head trauma to follow them as they cook and offer precise help on the recognized steps.

7.8.1 Contributions

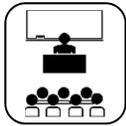
As stated in the introduction, the contributions of this paper are three-folds. The first contribution is practical and consists of an extensive dataset collected from a real smart home setup. The DOMUS laboratory is a real and complete smart apartment where people could live. It possesses everything we expect to find in a housing. While collecting the RFID readings, there were real obstacles between the bottle and the readers, like chairs around the table and a kettle on the counter. A computer simulation cannot reproduce those readings.

The second contribution concerns our methodology. While many researchers focus their work on the algorithm, we instead tried to model the problem differently and apply smart filters. As the first part of this work shows, most algorithms perform well and thus the learning part is not what has the biggest impact on tracking. To the best of our knowledge, few researchers tried to consider the localization and the tracking problems as classification problems. This qualitative localization seems less precise than a Cartesian localization. However, as shown in [20], qualitative spatial information is easier to use for ADL recognition by expert systems in smart homes.

The third contribution comes from our tracking experiment, where our realistic paths showed some theoretical limit to our localization data. Still, those limits formed recognizable patterns that we could detect and correct by designing some simple filters.

7.9 Acknowledgements

We thank the INTER research group and the FRQNT for their financial support.



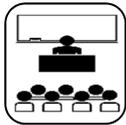
HUITIÈME CHAPITRE

RFID indoor localization using statistical features



8.1 Related work

8.2 Methodology



8.3 Experiments and Results

8.4 Acknowledgement

Résumé

L'article présenté dans le chapitre précédent décrit la méthode ayant permis d'obtenir un algorithme de positionnement intérieur. La méthode, basée sur une forêt d'arbres décisionnels (*random forest*), reconnaît la zone dans laquelle se situe une puce RFID passive avec une précision variable selon la pièce. Le jeu de données amassé pour ce premier article consiste en 50 lectures dans chaque zone, prises à 200 ms d'intervalle. Dans un article [27] publié à la suite de [14] (un article de conférence dont les thèmes sont repris dans l'article du chapitre 7), Bouchard critique la taille du jeu de données comme étant une limite pour l'utilisation de certaines méthodes. Il poursuit en montrant qu'en utilisant certaines mesures statistiques sur les données du jeu original, il est possible d'améliorer la performance de l'algorithme. Pour calculer les mesures statistiques, il considère alors les lectures de force du signal RFID comme étant des séries temporelles.

L'article « RFID indoor localization using statistical features » [17] est donc une réponse à [27]. Un nouveau jeu de données contenant 1 000 lectures pour chaque zone de la cuisine y est introduit. La méthode présentée par Bouchard est ensuite utilisée sur ces nouvelles données pour obtenir des résultats comparables. La contribution la plus intéressante de cet article, du point de vue d'une utilisation pour le BMP, est l'analyse de l'apport des différentes mesures statistiques à la précision du système. Il est ainsi montré que considérer moins de mesures statistiques ne nuit pas à la précision. L'algorithme ainsi dérivé est plus rapide que l'original de Bouchard. En comparaison avec l'algorithme présenté dans « Tracking objects within a smart home » [16], l'algorithme est légèrement plus lent, mais présente une précision de près de 11% plus élevée. De plus, les mesures les plus utiles sont aussi les plus simples à calculer, ce qui s'adapte bien à une utilisation sur un ordinateur de faible puissance.

Commentaires

L'article a été soumis et accepté pour une publication dans le journal *Cybernetics and Systems* publié par Taylor & Francis. Cet article est une réponse à un article individuel de Kevin Bouchard, qui est aussi coauteur sur cet article. Il a fourni le code permettant de calculer les mesures statistiques dont il est question dans l'article. Le candidat a ensuite pris ce code pour l'adapter aux nouvelles mesures présentées dans cet article. Le candidat a aussi fait toute la nouvelle collecte de données et la rédaction seul. Les coauteurs ont aidé à la révision de l'article pour en faciliter la lecture.

Cette thèse est au sujet du protocole de communication BMP. Pour fonctionner, le BMP exige que les objets en communication connaissent leur position dans l'environnement dans lequel ils sont. Cet article améliore la technique de positionnement décrite dans le chapitre 7. Ainsi, il contribue à améliorer la qualité des positions utilisables par un BMP.

Source originale

Titre de l'article : « RFID indoor localization using statistical features »

Auteurs :

Frédéric Bergeron, Département d'informatique, Université de Sherbrooke, Sherbrooke, Québec, Canada J1K 2R1, frederic.bergeron2@usherbrooke.ca

Kevin Bouchard, Département d'informatique et mathématique, Université du Québec à Chicoutimi, Chicoutimi, Québec, Canada G7H 2B1, kevin_bouchard@uqac.ca

Sebastien Gaboury, Département d'informatique et mathématique, Université du Québec à Chicoutimi, Chicoutimi, Québec, Canada G7H 2B1, sebastien_gaboury@uqac.ca

Sylvain Giroux, Département d'informatique, Université de Sherbrooke, Sherbrooke, Québec, Canada J1K 2R1, sylvain.giroux@usherbrooke.ca

Keywords

Indoor localization; passive RFID; data mining; feature extraction.

Abstract

In this paper, we present a method that uses the signal strength indication of RFID antennas with statistical features to perform relative positioning in a smart home. The goal of the proposed method is to enable the tracking of most objects inside a smart home in real-time, allowing activity recognition based on this tracking. This paper also introduces a new dataset of 4 100 000 RFID data collected in a real full-scale smart home setting. The dataset is available for the community. The method has an accuracy of 95.5% which is similar to previous work but require a fifth of the time to compute.

World population ageing is a situation most government are now fully aware they must face in the following years [119]. Related challenges are diverse, and so are the solutions proposed by researchers around the world. The challenges our teams are working on link to the increasing difficulty to provide adequate healthcare services to the ageing population. As life expectancy has increased in recent years, so have the number of persons suffering from age related diseases like dementia and Alzheimer's disease. This put healthcare systems under great pressure, both financially and in human resources.

Recent progress in miniaturization, especially for the Internet of Things (IoT), and the evolution of artificial intelligence algorithms have opened the way to the realization of the long-time dream of a real assistive smart home. From a healthcare point of view, a smart home is a regular housing that offers assistance in the realization of activities of daily living (ADLs). The goal of this assistance is to enable ageing in place while preserving the quality of life of all inhabitants of the smart home, thus delaying hospitalization and reducing the pressure on healthcare facilities and informal caregivers [11]. Indeed, the loss of autonomy of the elders often leads to long-term care in senior housing for non-autonomous or recurring hospitalization. The assistance can take many forms, from enhancing security of certain task like cooking to reducing the number of interventions needed from a relative living with an impaired person.

To replace effectively a long-term facility and to prevent accidents leading to hospitalization, a smart home must offer a large array of services and dispose of extensive data on its occupant [34]. However, collecting data on people is always a delicate subject as privacy concerns rapidly arises. To mitigate this problem, smart homes tend to avoid usage of video camera and instead rely on low level information collected using ubiquitous devices and technologies like passive infrared sensors, electromagnetic contacts, thermometers, wearable sensors and so on [34, 73, 12]. The upcoming field of IoT is regularly adding more low-level sensors a smart home can use. All those sensors provide useful data used to monitor ADLs in real-time [88]. Smart homes are also safer if they work on a private network, preventing access to cloud computing and forcing efficient algorithms that can execute on regular computers.

Still, data from unobtrusive low-level sensors are often not enough to provide precise information about what is going on, at least in their raw form. While they might be enough to determine that someone is cooking, they generally cannot tell what meal is the person preparing nor at what step a person is in a recipe. Several methods have been proposed over the past decade to recognize ongoing ADLs, but this endeavour remains problematic due to the low granularity of the current solutions. The granularity, in activity recognition, refers to the level of abstraction provided by the method. For instance, from the lowest to the highest granularity, the same ongoing ADL could be defined as: *Cooking*, *Preparing pasta*, *Preparing shrimp fettucine Alfredo*, or even as the atomic step *Putting fettucine in the boiling water*. While our teams at the LIARA laboratory and the DOMUS laboratory are fairly sensors agnostic [22, 123], we believe that one of the solutions with the highest potential to solve this granularity problem is the passive Radio-Frequency Identification (RFID) technology. The main advantage of passive RFID is that several tags can be installed on daily usage objects in the smart home to enable their tracking in real-time. Therefore, such system could provide highly reliable spatial information to feed an activity recognition algorithm for better granularity. As passive RFID tags are small and cheap, they can be placed on most objects. Nevertheless, RFID tags are not suitable to place on perishables or and cannot survive microwave oven nor high temperature.

In this paper, a localization system based on techniques for machine learning/data mining is proposed. The method builds upon the work of Bouchard [27] and Bergeron et al. [16] which is, in our knowledge, the only example of localization of several objects based on supervised data mining. Indeed, very few authors have worked on the problem of localizing daily usage object, and unfortunately, the best methods for humans/robots tracking often cannot be used straightforwardly because the technology used is too big (require batteries, antennas on the objects, etc.), is too costly, or requires several references points (disposing those in a smart home is not always feasible).

As it will be argued further in the paper, daily objects localization is more challenging than human or robot tracking, and the accuracy and precision of the state-of-the-art is still arbitrary. To address this challenge, in this paper, the RFID Received Signal Strength Indication (RSSI) is viewed as a time-series, as they were in [27]. In the aforementioned paper, Bouchard identified the low sampling of the dataset from [16] as a limit for a good evaluation of the proposed method. A new larger dataset is introduced to remove this limit. This paper pursue with the same research question Bouchard expressed: “How useful at improving RFID localization methods would be the statistical features commonly used in machine learning?”. Then, we follow with a new research question: “What statistical features are significant in improving our RFID localization method”. Accordingly, the contributions to the field of this paper, as an extension to [27], are three-fold :

- A new RFID dataset
- An improved pipeline for passive RFID localization
- A feature selection to improve computing efficiency

The dataset used in this paper is a new dataset of 4 100 000 RFID readings generated from real data collected in full-scale kitchen infrastructures and is available to the scientific community at www.kevin-bouchard.ca and www.usherbrooke.ca/domus. The previous dataset only contained 673 000 data dispersed in six rooms. The remainder of this paper is as follow. We first begin with a quick review of indoor localization methods, then we present of experimental settings along with the new dataset. After, we present the three series of experiment we conducted with their conclusions.

8.1 Related work

Research on indoor localization has been active for more than two decades [71]. Over the years, many approaches were created and tested for many different intended usages. A review of some usages can be found in [124], along with some challenges localization still poses before the emergence of a smart world.

Methods using wireless technologies can be regrouped in three main categories. A first category concerns the proximity-based methods. In proximity-based methods, we use the known position of fixed objects or tags to infer the position of the target object. The fixed objects can be wireless antennas, like with NFC localization [108], or tags, like in the LANDMARC system [114]. Considering the low range of NFC readers, the global idea is to say the position of the NFC tag is the same as the reader. For reference system like LANDMARC, the strongest signal among the references determines the position. Statistical features can help improve proximity-based method, as in this work [29] where the standard deviation of Bluetooth RSSI is used.

The second category are the lateration techniques that uses geometric properties to localize the target. Wireless antennas usually provide two types of information we can use for localization: the signal strength (the RSSI) and the angle of arrival (the AoA). Trilateration is the most used lateration technique using RSSI. The idea is to map the RSSI to a distance measure from the antennas and draw virtual ellipsoid to pinpoint the location at the intersection of few reference points [66]. On the other hand, triangulation is the most popular method using the AoA. Instead of drawing virtual ellipsoid, triangulation demands to draw virtual straight lines and places the target at the intersection of at least two of them [147].

The last family and the one of interest for this paper is the fingerprinting. The fingerprinting technique is usually used in conjunction with a better, more precise, localization system to build a radio map of the environment. The technique is, then, to use the learned map and compare, in real-time, the RSSI to associate the tracked entity to the closest location in a similar fashion than with landmarks. The main drawback is, however, the requirement for the high-performance localization system (usually based on ultrasonic sensors) [72]. The more precise system can be replaced by a manual collection of the fingerprints [16]. Nevertheless, manual collection is a long and tedious process. Still, it allows doing relative positioning and varying the precision at will. Fingerprints also allow using less antennas than lateration technique as the target only need to be in the range of a single antenna for the method to work. Fingerprinting are also used with non-wireless technologies. Those includes sounds, magnetic fields and, to some extent, light. SurroundSense [2], for instance, is a user-centred fingerprinting localization application that uses the user's cellphone to capture the photo-acoustic signature of a place to later provide localization. SurroundSense also fingerprints the motion using accelerometers, the colour using the camera and the Wi-Fi, when available, to provide better localization. In their paper, the authors tried to localize in what shop the users were between 51 shops. LocateMe [140] is another user-centred localization method. It uses the magnetometer of a cellphone to record the ambient magnetic field (a combination of the magnetic field of the Earth and the local distortions provoked by metallic structures) in hallways. Then it uses dynamic time warping to match ongoing a user's location to a previously recorded fingerprint. By using dynamic time warping, LocateMe can adjust to various users with various walking speed or disabilities like blindness and paralysis requiring a wheelchair. However, those systems mainly use a cellphone that as to be carried by a user for the localization to occur. In their current form, the technologies they use cannot be placed on daily living objects. Moreover, LocateMe computes the position directly on the cellphone, in about 5 seconds. To our sense, this is too slow to use for real-time activity recognition, our final goal.

8.2 Methodology

In this section, the goal is to explain the methodology used to validate the research questions formulated in the introduction. While the emphasis of this paper is on the localization of one object in one smart home, the reader should keep in mind the bigger picture, which is about tracking several objects in real-time for ADLs recognition in smart homes to foster aging at home. Our team has already used the spatial data from passive RFID localization in activity recognition system in the past [13] and improvements in the localization tend to translate directly in better activity recognition.

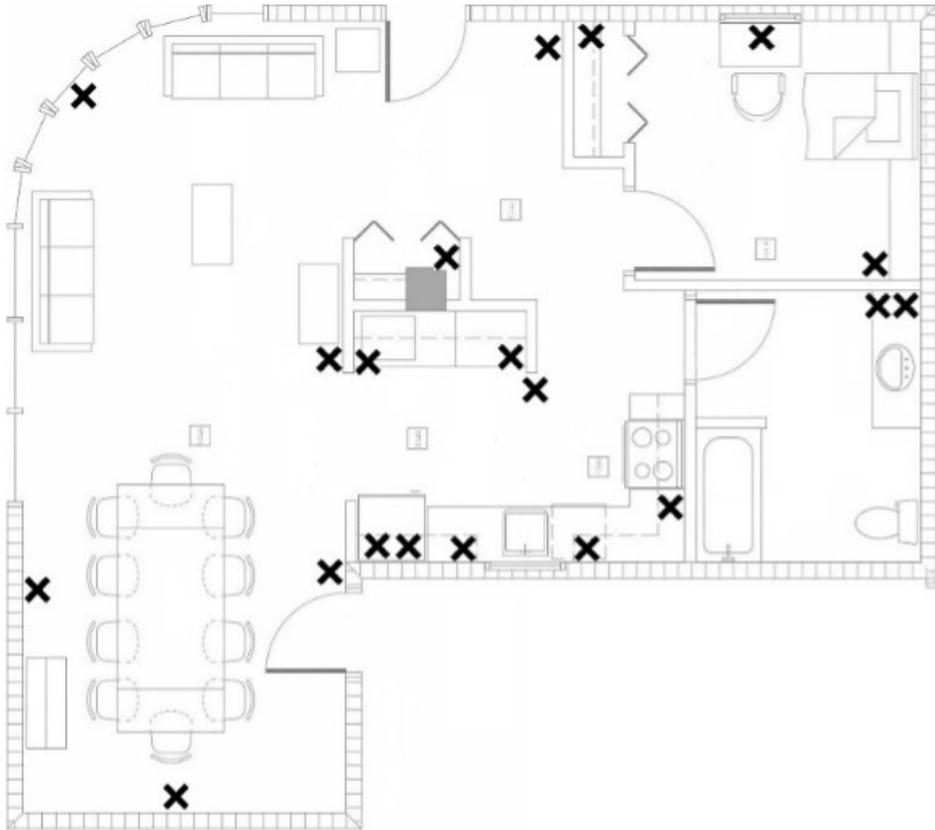
This paper is an update on the work of Bergeron et al. [16] conducted with the LIARA and the DOMUS teams from which the author are members. The method exploited in the aforementioned paper relied, similarly to the literature, on using the raw RSSI signal from the passive tags to perform the localization. This paper is also a follow up to the work of Bouchard [27]. In contrast, Bouchard, in his project, sees the RSSI as a time-series. Therefore, despite the low sampling of the dataset he used, the localization is performed over a data window, which is an aggregation of many consecutive readings. The importance of this work relies on the premise that daily objects localization is more difficult than human/robot localization. The arguments are that daily objects can be very small (e.g.: a spoon, or a fork), numerous (in the kitchen there are several plates, containers, glasses, etc.) and that occlusion will often occur. Small objects implies that the tracking technology also have to be small and light. Numerous objects in turn implies that the localization process needs to be as efficient as possible in order to resolve in real-time and to scale graciously as more objects are added. Frequent occlusion disallows the use of line-of-sight technologies. It also requires the radio signal to operate at a frequency where interference with the environment are minimal. A smart home is hardly considered an open environment and thus the challenges of localization are different. The method we propose with this work tackles those three challenges by using passive RFID tags with an efficient localization method described below.

8.2.1 Smart home

In the introduction, we mentioned that a major contribution from this paper is a large dataset of RFID readings collected in a realistic smart home setting [16]. This subsection will delve into the smart home and the next will present in more details the new features of this dataset. The smart home is a full-scale apartment including a bedroom, a kitchen, a dining room, a living room and a bathroom. It is equipped with 20 polarized directional antennas distributed to cover the entire surface, as shown in Figure 8.1. These antennas are connected to five RFID readers and work on the 928Mhz band as specified by the Canadian Radio-television and Telecommunications Commission (CRTC). Therefore, they have to be strategically installed to minimize collisions and maximize coverage. Collisions cannot occur among the antennas connected to the same reader since they work on a round robin. A derogation can often be obtained through the CRTC to change the band, but since our goal is to use the smart homes for aging in place, this would not be practical. In theory, the RFID system can collect the tags ID up to every 20ms. However, RFID is not a real-time system, thus the results are often very different. In practice, it has been observed to be reliably able to collect data under 100ms. They were set to collect at every 200ms for this experiment. The smart home features many more sensors, like pressure plates, infrared detectors and thermometer. For this paper, only data from RFID antennas are used. However, all other data sources were active during that time and might have caused some ambient noise. Still, this noise is expected to be present in a real-life smart home and is accordingly considered beneficial to this work.

Figure 8.1

Map of the smart home and position of the RFID antennas



8.2.2 Dataset

The smart home presented previously offers a surface of about 100 square meters. As explained in previous papers [16, 27], we consider the localization problem as a classification problem where positions are given in zones rather than in coordinates from an arbitrary origin. As for most classification problems, we first need to collect a dataset to train a classifier. The smart home being quite large, we choose to only collect RFID readings in the kitchen for the current dataset, as opposed to the six rooms of the previous dataset. To better compare with the previous articles, we used the same 205 zones from the kitchen as before. We also used the same plastic bottle with the same four RFID tags facing different directions. What changed for this experiment was the number of collected readings and the state of the smart home. Indeed, this time we collected 1 000 readings for each zone (it was only 50 per zone in the previous dataset). At 200ms between each reading, it took 3 minutes and 20 seconds per zone. During that time, we tried to mimic usual activities performed in any kitchen, like cleaning the dishes or cooking. Whereas in the first dataset readings were collected under perfect conditions with no human caused interference, this dataset contains normal interference we expect to find in an inhabited smart home. Therefore, 1 000 readings per class for each antenna were recorded resulting in 205 000 vectors of twenty RSSI + one class or 4 100 000 data. The variation of the RSSI values is bounded between -38 to -69 decibel. Due to the high number of readings per zone, some vectors appear more than once in the dataset. In total, there are 184 057 distinct vectors. This dataset will be exploited as is in the first experiment to present the new baseline for future comparisons.

8.2.3 Statistical features

Bouchard showed in [27] that statistical features extracted from the dataset collected in [16] can improve the classification accuracy. Since this new dataset is essentially an expanded version of the previous one, the hypothesis of the second experiment is that the features should also improve the classification accuracy on the new dataset.

The vectors contained in the dataset each represent the values at each antenna at a 200ms interval. They are regrouped to form a time-series. We refer to the size of the grouping as the data window. Then, on those time-series, it is possible to extract statistical features to take advantage of this group of readings. The features we used are presented in Table 8.1. The notation used considers M to be the matrix of the data window made of k lines and n columns (the antennas). There are nine statistical features applied to each time-series and eight applied globally (to all 20 time-series). For instance, the Mean RSSI is the sum of all RSSI in a window for an antenna divided by the window size. The Global Mean RSSI is the sum of all RSSI in that window divided by the total number of elements in that window ($n * k$). Consequently, the size of each features vector is 189 ($20_{time - series} * 9_{statisticalfeatures} + 8_{globalfeatures} + 1_{class} = 189$).

Table 8.1

The list of features applied on the RFID readings

$\bar{x}_j = \frac{\sum_{i=1}^k x_{i,j}}{k}$	$GAvg = \frac{\sum_{i=1}^k \sum_{j=1}^n x_{i,j}}{n*k}$	$\min(j) = \text{Min}_{i,t}\{x_{k,j}\}$
$Var(X_j) = \frac{\sum_{i=1}^k (x_{i,j} - \bar{x}_j)^2}{k}$	$\sigma_j = \sqrt{Var(X_j)}$	$GMIn = \min_n\{\min(j)\}$
$NZ_j = 1 - \delta_{x_{i,j}}$	$GStDev = \frac{\sum_{j=1}^n \sigma_j}{n}$	$\max(j) = \max_t\{x_{k,j}\}$
$GNZ = \sum_{j=1}^n NZ_j$	$E_j = \sum_{i=1}^k x_{i,j}^2$	$GM ax = \max_n\{\max(j)\}$
$SC_j = \sum_{i=1}^n x_{i,j} - x_{i-1,j} $	$GE = \sum_{j=1}^n E_j$	$MCj = \frac{\sum_{i=1}^k x_{i,j} - x_{i-1,j}}{k}$
$GSC = \sum_{j=1}^n SC_j$	$Tp = \sum_{i=1}^k \sum_{j=1}^n x_{i,j}$	

Most of these features are common knowledge, but few of them may need a proper introduction. The Count Non-Zero, and by extend the Global NZ, count the number of occurrences where a signal was read, or to simply put where the RSSI was different from zero. It is expressed using the Kronecker delta. The Absolute Energy is the sum over the squared RSSI values. The Mean RSSI Change is the average fluctuation in RSSI that can be expected on the time-series. The Absolute Sum of Changes (and Global SC) is the sum over the absolute difference between each consecutive RSSI values. Finally, the Global Total Power is the sum of all RSSI values over each time-series of the window.

There are many more features in the literature that we could have used. Features based on Fourier transform, for instance, are popular to work on time series. However, they require expensive mathematical computation that might not scale well for a real time usage on many objects. In fact, this real-time constraint prevents us to use more complex features than those presented above, even though some of them might improve the accuracy of our classifiers.

8.3 Experiments and results

Three experiments are presented in this section. The first reprises the classification work from [16] using the new dataset. The second reprises the work from [27] also using this new dataset. The last one goes further on the statistical features by examining which are the most used. The third experiment, along with the new dataset, is considered the main contribution of this article.

8.3.1 Raw readings classification

The goal of the first of the three experiments is to establish a new baseline for classification on the new dataset. To do so, we used the same protocol we used before. We use classification algorithms from the popular Waikato Environment for Knowledge Analysis [157], Weka, on the dataset without any pre-treatment using only default parameter. For this part, the goal is not to find what parameters combination gives the highest accuracy. Instead, the goal is to provide a rough preview of what can be achieved by common classifier on this dataset for the task of indoor localization. The results presented in this section were all obtained using default parameters and 10-fold cross validation as provided in Weka. Table 8.2 presents the accuracy for a selection of algorithms and compare them to the results presented in [14] for the kitchen.

Table 8.2

Accuracy of some classification algorithms on the raw dataset

Algorithm	Previous dataset	New dataset	Diff +/-
CART	73.403%	80.601	7.198
J48 (C4.5)	74.966%	80.269%	5.303%
Random tree	67.815%	75.696%	7.881%
Random Forest	88.916%	86.590%	-2.236%
Naive Bayes	83.227%	46.054%	-37.173%
1-NN	78.824%	82.299%	3.475%

Results show an increase of the accuracy for most algorithms. The random forest performs a little worse than Bergeron & al. [14], but it is still the best algorithm among those we tried. The weighted F-Measure for all algorithms is the same as the accuracy. For J48, it is 0.8077 and for CART it is 0.8060. In all cases, there is still a need for improvement, as being wrong 15% of the time is unpractical in order to build a reliable ADLs recognition system using this method. Given there are twenty times more training examples in the new dataset, an increase in accuracy was expected. With more example, we are closer to have all possible values seen while training. This also means that the goal is not to have a classifier good at generalizing but a classifier that can remember well. Trees are good in this context and it reflects in the accuracy.

8.3.2 Statistical features classification

The first experiment showed that using a larger dataset allows an increase in the accuracy. However, this amelioration is not enough to build a robust ADLs recognition system on top of it. As Bouchard showed in [27], statistical features extracted from a time series formed by the RFID vectors contained in the dataset can improve the accuracy of the indoor localization while retaining real-time capacities. The second experiment of this paper simply pick up the work of Bouchard and applies it to the new larger and noisier dataset.

Table 8.3 presents results with a data window of five. The accuracy of the various algorithms is similar to those obtained by Bouchard. Considering that the new dataset contains noise, a small decrease in the accuracy was to be expected. The difference in accuracy between the feature dataset and the raw dataset is less marked than in [27], but since the baseline accuracy is higher than before, this too was to be expected. The weighted F-Measure of those algorithms reflects the accuracy. For instances, the weighted F-Measure of J48 is 0.9537 and is 0.8404 for the nearest-neighbour algorithm (1-NN).

Table 8.3

Accuracy on the feature dataset with a window of 5

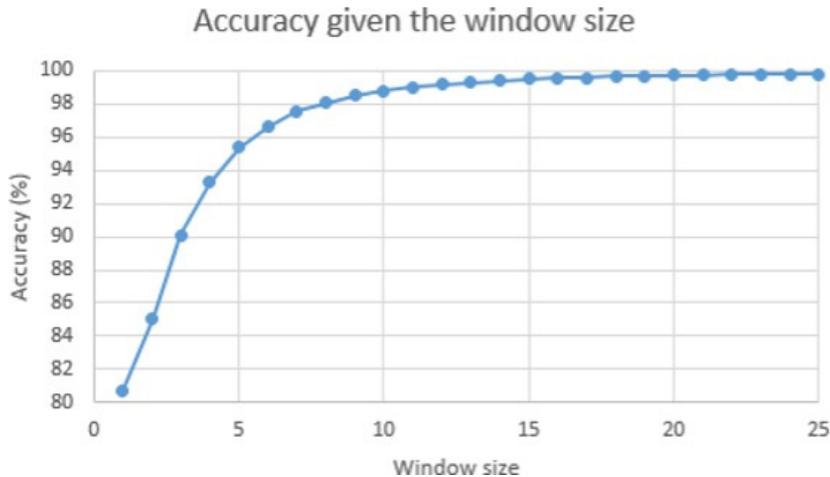
Algorithm	Raw dataset	Feature dataset	Diff to raw +/-	Diff to Bouchard
CART	80.601	94.437%	13.836%	0.437%
J48 (C4.5)	80.269%	95.378%	15.109%	-1.022%
Random tree	75.696%	83.649%	7.953%	-5.951%
Random Forest	86.590%	98.232%	11.642%	-1.568%
(100)	46.054%	70.670%	70.670%	24.616%
Naive Bayes	82.299%	83.995%	1.696%	15.495%
1-NN				

8.3.3 The impact of windowing

In his paper, Bouchard also explored the impact of windowing. His conclusion was that accuracy improves with the window size with a saturation around 99.53%. We also reproduced this experiment with a J48 classifier and the results are given in Figure 8.2. With a window size of one, the accuracy is 80,714%, which is slightly higher than the accuracy on the raw dataset. The accuracy never ceases to improve within the selected boundaries. At size 25, the accuracy is 99,813%, meaning that only 375 examples are misclassified. If accuracy was the only criterion, this window size would be a good choice. However, a window size that big is not practical for real-time positioning, as most moving objects would cross multiple zones during the five seconds needed to record 25 readings. This is unpractical for online real-time tracking. A window of five readings, at 95,370% accuracy, seems a good trade-off between accuracy and practicability when the delay between two consecutive RFID readings is 200ms. The choice of the window should reflect both the RFID sampling rate and the speed at which objects are expected to move. If objects are expected to be fixed, like in a warehouse, a large window will offer a better accuracy. For activity recognition based on object movement, a smaller window allows a more precise tracking, especially while a tracked object is being used. With the size of our zones (20 cm), we expect objects to be moving at a speed between two to five zones per second.

Figure 8.2

Accuracy of the J48 classifier for a window size varying between 1 and 25 RFID vectors.



Apart from the accuracy, another impact of the windowing is the size of the produced tree. A window of 5 results in a tree of 10 563 nodes, while a window of 10 produces 4 667 nodes and a window of 25 results in a J48 tree of 1 965 nodes (of which 848 are leaves). This suggests that most zones have a distinct signature that is easier to detect when more readings are aggregated. This also suggests that some features might not have a significant impact in the classification as they have less to gain from more data. The max and the min RSSI are the features that have to most to gain from a bigger window as more data increase the chances of finding new extreme values. If there are no extreme values and instead they are similar for each vector, measures of central tendency should not be much impacted by more data. The next experiment aims to find what features are the most significant during the classification.

8.3.4 Selection of features

The previous experiments showed that the feature approach helps to improve the accuracy of classification algorithms on a noisy dataset of RFID readings collected from a real-life smart home. The goal of the indoor localization system presented here is to build an ADLs recognition system based on the tracking of objects. This means that for real-life settings, the system will need to track near hundred of objects in real-time. At this scale, every computation becomes significant. With that thought in mind, we used several methods to try to reduce the number of features. There are 9 statistical features and 8 global features applied on each sliding window.

We first tried the principal components analysis (PCA) method as a wrapper over a J48 tree. The result was a tree of more than 40 000 nodes. PCA formed 25 new features using most of the original ones. In fact, it used at least once every statistical feature, meaning we could not remove any of them. We then used the information gain wrapper retaining only the features that offered a gain ratio superior to zero. There were 80 features that qualified this criterion, once again including all the original features on at least one antenna. The accuracy is also similar to the original feature dataset, with 95.504% on a tree of 10 559 nodes. We then tried to limit the number of original attributes used and the number of produced features, only to obtain similar results.

Since the classical feature selection algorithms were of limited use in our context, we had to think of a less mathematical method. We designed the following experiment to find what features are the most used and what is the impact of removing a feature, both in time and in accuracy. First, we take all features and train a J48 tree on it. Then, we list the frequency of all features used in the tree. Finally, we remove the least used feature and go back to training a new tree. We repeat this process until there is only one feature left in the dataset. For the statistical features, we count them for all antennas. Therefore, if the Min RSSI is used 10 times for antenna 4 and 8 times for antenna 6, the count is 18 for the feature. To us, it made no sense to remove a feature only for a given antenna, as it would not bring any knowledge transferable to another room or another smart home setting.

In Table 8.4, we listed the accuracy resulting at each step of our removal method. We also list the number of nodes in the resulting tree to give an idea of the complexity and execution time of the resulting tree. The computation time is the time it took to compute the features on the dataset, as this was done in an off-line phase. Those times should be considered carefully, as they are only an indication of the computation complexity of computing the features. This task was done on one of our work computers, while other programs were also running. Therefore, they are only given so we can appreciate the difference between the features. The last columns indicate how many features were used in the classification tree versus how many were present at this stage of the experiment. Given that the RFID antennas are placed to cover the whole smart home and not only the kitchen, some of them never recorded any reading. Therefore, it is normal to see that they were not used in the tree as we can see in the last column.

Table 8.4

Accuracy of J48 when removing features with window of 5

Feature removed	Computation time	Accuracy	Number of nodes	Features used
None	161s	95.445%	10563	67 o 18 f 8
Global Mean RSSI	157s	95.433%	10573	67 o 18 f 7
Global Abs Energy	160s	95.437%	10589	66 o 18 f 6
Standard Deviation	140s	95.457%	10577	66 o 16 f 6
Global SC	133s	95.480%	10553	65 o 16 f 5
Global Total Power	135s	95.500%	10557	65 o 16 f 4
Global Mean StDev	133s	95.490%	10581	64 o 16 f 3
Global Min RSSI	121s	95.474%	10555	63 o 16 f 2
Global Count RSSI	121s	95.504%	10549	61 o 16 f 1
Global Max RSSI	117s	95.135%	11029	60 o 16 f 0
Variance	96s	95.188%	11113	54 o 14 f 0
Mean Change	97s	95.322%	11023	46 o 12 f 0
Abs Sum of Changes	75s	95.471%	10919	38 o 10 f 0
Abs Energy	57s	95.532%	10935	31 o 80 f
Mean RSSI	46s	95.772%	11349	24 o 60 f
Count Non-Zero	41s	96.054%	10987	16 o 40 f
Min RSSI	35s	95.104%	12761	8 o 20 f

As we can see in the table, removing features does not have a big effect on the accuracy. In fact, the accuracy varies around 95.5% for most classifiers. The real difference between each row are the computation time and the number of nodes. In order to have the fastest system possible, both need to be minimized. However, the number of nodes generally grows as we remove features. Still, computing only the Max RSSI takes about a fifth of the time needed for all the features while producing a tree a fifth bigger. Nevertheless, since trees have a search complexity of $\log_2 n$, the increase in the size of the tree is not enough to slow the classification (it adds about 0.2 node in average). The conclusion of the experiment whose results are in Table 4 is that the most useful feature is the Max RSSI, when using a window of 5. With only this feature, we can build a classifier (J48) that is about as fast and as accurate as those presented in the second experiment. This feature is also the most significant when we use a window of 25, with an accuracy of 99.757% (although a window of 25 represents 5 seconds of data collection and can only be applied to fixed objects).

Table 8.5

Individual features with window of 5

Feature	Accuracy	F-Measure
Count Non zero	20.490%	N/A
Absolute Energy	84.877%	0.8494
Abs Sum of Change	19.358%	0.1906
Mean Change	12.861%	0.1255
Mean RSSI	88.617%	0.8860
Minimum	90.671%	0.9068
Maximum	95.104%	0.9568
Variance	28.988%	0.2885
Standard Deviation	29.090%	0.2894

Individual features

The iterative process presented above allows to isolate to most significant feature by removing a feature at the time. It shows that the max RSSI alone offers a good accuracy. Still, we decided to evaluate the other features to see if alone they can perform well. The results are presented in Table 8.5. From this table, it appears that the mean, the min and the max RSSI are the three features that contribute the most to the accuracy. This was to be expected since the min RSSI is the last feature removed in the previous experiment and the max RSSI is the remaining one.

Another J48 tree was trained on a dataset containing the four best features identified in Table 8.5 (absolute energy, mean RSSI, maximum RSSI and minimum RSSI). It outperforms most classifiers from Table 8.4 with an accuracy of 95.853% (8532 incorrectly classified instances). The weighted F-Measure is 0.958 and the tree has 10495 nodes in it. For a generation time of 78 seconds, it can be seen as maybe the best trade-off between accuracy, features computation time and execution time.

8.4 Conclusion

In this paper, we proposed an indoor positioning system based on passive RFID tags. We modelled the indoor positioning problem as a classification problem, where we aimed to find in which zone a tag was rather than at which coordinates. We considered RFID readings as a time series on which we computed several statistical features and built classifiers on those features. Then, we examined what features were the most useful in our context. Our results showed that using statistical features can help improve the localization accuracy by more than 15% when compared to using the raw RSSI readings. The time series were formed by aggregating several consecutive readings by using a sliding window.

The size of the window has an impact on the accuracy. With a window bigger of 25, the accuracy is more than 99.8%, but it is only usable with fixed objects as it takes 5 seconds to collect these readings. Our last experiment showed that simple statistical features are as useful as more complex ones. In fact, using only the max RSSI value of the window offers an accuracy comparable to using all the 17 features we first tried. Still, the absolute energy, the mean RSSI and the minimal RSSI values for each antenna also offer an increase in the accuracy compared with the raw dataset. Together, those four features create one of the best J48 tree of the whole experiment with an accuracy of 95.853%. Those are the features we intend to keep for our upcoming real-time activity recognition system.

All those experiments were made possible by the gathering of a new large dataset of RFID readings. The readings were collected in a full-scale smart home while daily living activities were performed and while other radio-frequency technologies (Wi-Fi, ZigBee, Z-Wave) were in use. This makes this dataset suitable for real-life usage as it already includes many natural forms of noise we expect to find in a smart home. The dataset of 4 100 000 data is freely available for all researchers to use. Free data in that quantity is hard to find as smart home settings are still uncommon and costly to build.

The indoor localization system presented offers an accuracy of about 95% with a window of 5 readings. However, our method requires a tedious work of fingerprinting the smart home that must be done for each new home. Future work could focus on automating this process or analyzing what precision is required for different categories of task to reduce the number of needed zones. A direct follow up of this work would be to use the system as a building block for a tracking system or for an indoor activity recognition system based on the position of objects. The localization system can also provide positioning for a large range of IoT applications. As the proposed method of statistical features only uses RSSI as an input, it should be replicable with other radio-frequency technologies. Future work might also wish to explore if those other technologies offer the same accuracy.

8.5 Acknowledgement

This work was made possible by the financial support of the Natural Sciences and Engineering Research Council of Canada.

« I know all about endings. It is beginnings that elude me. »

— Marion Zimmer Bradley, *Ancestors of Avalon*, 2004

CONCLUSION

L'histoire de l'humanité en est une de révolutions technologiques et de révolutions sociales. La séparation entre le paléolithique et le néolithique est technologique : il s'agit du développement de l'agriculture et de l'élevage organisé (vers 10 000 av. J.-C.). La séparation entre la préhistoire et l'histoire est aussi technologique ; cette fois, il s'agit de l'invention de l'écriture (vers 3500 av. J.-C.). Si traditionnellement notre époque contemporaine (occidentale) débute avec les révolutions américaines (1776) et françaises (1789), la révolution industrielle anglaise (vers 1760) est la révolution qui aura le plus marqué le monde en permettant la production de biens sur une échelle inimaginable à l'époque. La révolution industrielle aura modifié de façon permanente les relations entre les humains et le travail. Nous sommes à l'aube d'une autre révolution qui promet de modifier tout aussi durablement les relations entre l'humanité et la machine. La révolution informatique débute au cours de la seconde guerre mondiale avec les travaux de Turing et se poursuit dans les années 1970 avec les ordinateurs personnels. Depuis les années 2000, cette révolution connaît toutefois un second souffle avec les progrès de l'intelligence artificielle et l'apparition des objets intelligents grand public. Maintenant, ces objets intelligents sont devenus si nombreux et si sophistiqués qu'on y réfère comme étant des parties de l'Internet des objets.

Ce nouveau paradigme de l'informatique ubiquitaire fait toutefois face à de nombreux défis. Certains d'entre eux touchent à la communication entre les objets. Il existe de nombreux protocoles permettant la communication entre les objets. Une caractéristique commune relie tous les grands protocoles grand public : la nécessité de connaître avec qui il est possible de communiquer, pour former un réseau dans lequel chaque objet peut communiquer directement avec un autre objet.

Cette caractéristique est souvent utilisée afin de rassembler l'information collectée par les objets vers un serveur central, qui risque alors de se retrouver submergé par les données. Plutôt que de s'attaquer à trouver un meilleur algorithme de sélection des données importante, cette thèse cherche plutôt à présenter un nouveau modèle de conception pour les applications de l'IdO dans lequel les objets sont inconnus et la notion de serveur central est absente.

La thèse proposée dans ce document vise à pallier certains problèmes de communication auxquels fait face l'Internet des objets par l'élaboration d'un nouveau protocole de communication. Il est souhaité que ce nouveau protocole ait un impact positif sur la durée de vie des piles des objets en réduisant le nombre de messages échangés. Il est aussi prévu qu'il diminue la quantité d'informations en mémoire en ne conservant que l'essentiel.

Contributions en communication

Le protocole proposé se nomme BMP, pour *Bounded Message Protocol*. Son nom provient de sa caractéristique principale qui est de donner une limite physique de distance à franchir pour les messages. Cette limite agit donc comme une frontière (en anglais, *boundary*) au-delà de laquelle les messages BMP doivent être ignorés. Cette limite physique est la seule limite à la propagation d'un message dans un environnement. Dans le BMP, il n'y a pas de notion d'adresses d'origine et de destination. Il n'y a que des messages, envoyés depuis une position physique connue. Ces messages sont envoyés à tous les autres objets capables de recevoir le signal portant le message. À la réception, ces objets ont alors la tâche de déterminer s'ils respectent ou non la distance en connaissant leur propre position.

Conclusion

Les travaux reliés au protocole BMP sont présentés dans la deuxième partie de la thèse. Dans un premier chapitre, les aspects conceptuels du protocole sont introduits. En plus de la notion de position donnant une portée maximale à un message, on y présente la limite temporelle qui accompagne les messages BMP. Cette limite indique le temps qu'un message devrait être conservé après sa réception. Cela confère une durée de vie aux messages permettant de faire en sorte que seuls les messages encore valides soient enregistrés sur un objet, permettant ainsi de réduire le problème de la quantité de données à traiter qui a été identifié plus tôt. Si le temps est préférablement traité en seconde, les positions et les distances ne supposent aucun référentiel. C'est à l'utilisateur du protocole d'implémenter son propre module de traitement des distances. Deux exemples de modules sont présentés dans la thèse. Un premier module utilise les coordonnées géographiques habituelles et exprime les distances en mètres. Un second module représente les positions et les distances comme des mots, agissant ainsi à la façon d'un système producteur-consommateur.

Dans un deuxième chapitre, les aspects d'implémentation du BMP sont présentés. On y détaille d'abord un diagramme décisionnel dans lequel on retrouve les grandes étapes de traitement d'un message entrant. Les étapes sont ordonnées pour viser l'efficacité de l'implémentation. La première étape consiste donc à vérifier si le message est déjà en mémoire. S'il n'y est pas, on évalue ensuite les limites temporelles et physiques. L'évaluation de la distance physique peut être longue à accomplir, c'est donc une étape critique pour la performance. Le BMP a été implémenté pour deux types d'objets. Une première implémentation est en C++ et vise les objets Arduino. Les étapes du schéma y sont implémentées sous la forme d'une machine à états afin de pouvoir continuer à recevoir des messages pendant le traitement d'un autre message. Ce mode de fonctionnement est idéal pour les systèmes n'ayant qu'un seul fil d'exécution. Une seconde implémentation est en Java et vise la plateforme Android. Android supportant des fils d'exécution multiples, l'implémentation Java en fait usage.

Un troisième chapitre sur le BMP rassemble les résultats découlant des déploiements des diverses implémentations sur de vrais objets. Un premier déploiement a été fait sur des objets DecaDuino (de la famille Arduino) à l'IUT de Blagnac. De ce déploiement, on vérifie que le mode de propagation basé sur la distance entre les objets fonctionne. On y vérifie aussi que l'implémentation est suffisamment efficace pour permettre une communication entre plusieurs objets, qualitativement. La communication s'y fait par UWB. Un deuxième déploiement a lieu du DOMUS de l'Université de Sherbrooke, cette fois sur un mélange de tablettes et de téléphones Android. L'expérience consiste cette fois à produire un scénario de déplacement et d'activation de capteurs plausible pour une maison intelligente. La fréquence d'émission des messages sur les différents objets est toutefois plus rapide que dans la réalité, afin de réduire la durée du scénario et de provoquer une charge plus grande pour le BMP par la même occasion. Cette expérience présentant des résultats quantitatifs, on remarque que les 10 objets présents dans le scénario reçoivent entre 90% et 95% des messages au moins une fois, ce qui est jugé suffisant pour cet environnement. On constate toutefois que les temps de traitement des messages varient entre 200 ms et près de 600 ms, ce qui est plutôt lent en plus d'être d'une grande variance. Ces temps de traitement peuvent laisser croire que des messages sont peut-être perdus par un trop grand délai entre deux lectures sur la radio Wi-Fi. Il serait intéressant de refaire l'expérience avec des objets plus modernes que les vieillissantes tablettes utilisées.

Certes, une étude en simulation du protocole aurait pu remplacer ou être intégrée à ce troisième chapitre pour vérifier de manière complémentaire les objectifs de notre thèse. Toutefois, la simulation de protocoles et de réseaux présente des limitations [130, 86, 100]. Aussi nous avons choisi de ne pas retenir cette voie malgré les risques et aléas inhérents à une approche de vérification en milieu physique complexe réel, ce qui a été confirmé lors de nos expérimentations comme nous l'avons souligné lors de nos discussions dans cette thèse. Une étude en simulateur serait tout de même intéressante dans le futur pour évaluer le protocole plus rapidement dans diverses configurations.

Conclusion

Le dernier chapitre dans cette partie sur le BMP vient apporter différentes intuitions sur l'utilisation future du protocole. On y spécifie tout d'abord que le BMP vise à être utilisé par les systèmes à base de règle où l'information est accessible à tous et est disponible dans l'environnement rapproché. On y précise que le BMP n'offre aucune garantie de livraison des messages, élément devant être pris en compte lors de l'élaboration d'application voulant l'utiliser. Deux exemples de telles applications sont décrits en détail dans le chapitre. D'abord, on présente comment le BMP peut déjà être utilisé pour développer une application de soutien à domicile pour le problème d'errance nocturne que souffrent parfois les gens atteints de la maladie d'Alzheimer. Ensuite, dans une application futuriste, le BMP est utilisé pour concevoir une intersection intelligente où la signalisation est réduite à un minimum. Les voitures y sont toutes autonomes et communiquent avec les différents éléments de l'intersection pour savoir où aller et quand elles peuvent le faire en toute sécurité. Ce scénario est propice à faire ressortir les défis apportés par l'absence de garantie de livraison des messages. Cet élément stimule l'introduction d'une redondance pour les messages les plus critiques.

Cette thèse se consacre principalement à présenter le nouveau protocole de communication BMP. De nombreux autres travaux restent à faire sur ce protocole pour mieux en comprendre les capacités et limitations. Un premier travail futur déjà identifié au chapitre 5 est d'établir les conditions limites propres à chaque implémentation. Il est principalement important de définir la relation entre le nombre d'objets participants à la communication et la fréquence des messages conduisant à un effondrement des performances du système. En effet, puisque chaque message envoyé est répété par tous les objets acceptant ce message, une grande densité devrait conduire à une grande charge sur le médium utilisé pour la communication.

Sur le front de l'utilisation du BMP, en plus des exemples d'applications déjà présentés, il sera intéressant d'examiner si le BMP n'ouvre pas la voie à une nouvelle génération d'algorithmes faisant de la position des choses un élément aussi central que dans le protocole. On peut, entre autres, penser à des algorithmes de fusion progressive d'information à mesure qu'on s'éloigne de la source d'un message.

Concernant le développement futur du BMP, de nombreuses améliorations sont aussi possibles. La gestion des positions et des distances est actuellement plutôt rigide dans le sens où la taille allouée pour ces éléments dans l'en-tête des messages est fixe. Il serait intéressant de rendre ces champs de taille variables et d'en mesurer les impacts. Le BMP n'inclut actuellement aucune notion de sécurité, préférant laisser cette tâche aux applications l'utilisant. Il serait toutefois possible d'ajouter une forme de sécurité basée sur la chaîne de bloc. Les contrats intelligents [35] sont un exemple d'une façon de permettre l'interaction sécuritaire entre objets ne se connaissant pas. Puisqu'aucun objet ne se connaît dans un BMP, il s'agit d'un domaine d'intérêt tout indiqué pour explorer cette technologie.

Contributions en positionnement d'objets

La dernière partie de cette thèse se concentre sur les contributions en positionnement réalisées pendant l'élaboration du BMP. Puisque la position est un élément central du BMP, il importait d'avoir un mode de positionnement intérieur pour objets facilement réalisable dans le contexte du DOMUS. L'approche choisie est de considérer le problème de positionnement comme étant un problème de classification. On cherche donc à classer un vecteur contenant la force d'un signal RFID entre une puce collée à un objet et les 20 antennes RFID du DOMUS. Les classes possibles sont des zones de taille variable où des vecteurs de référence ont été obtenus. La méthode produite est nommée *Indoor Positioning for Activities of Daily Living*, IPADL.

Conclusion

Le premier chapitre de cette dernière partie de la thèse présente la construction de la méthode de positionnement, en présentant tous les classifieurs utilisés. Il en ressort que la forêt d'arbres de décision aléatoire est la méthode la plus rapide présentant un bon taux de succès dans le positionnement. Ce même chapitre poursuit ensuite en présentant divers filtres créés afin de stabiliser les mesures de position sur un objet en déplacement. En effet, les arbres de décision utilisés n'incluent pas la notion de proximité physique entre les classes. Ainsi, une mauvaise identification conclut souvent que l'objet est maintenant dans une position diamétralement opposée à sa position précédente, ce qui n'est pas souhaitable. Les filtres viennent alors limiter le déplacement des objets à une case à la fois, dans la direction de la nouvelle position. Le deuxième et dernier chapitre de la seconde partie de la thèse présente une amélioration de la méthode IPADL. L'amélioration principale consiste à considérer les vecteurs de force de signal successifs comme des éléments d'une série temporelle. Il devient alors possible de calculer des mesures statistiques sur ces forces pour les ajouter au vecteur à classifier. La méthode est originellement conçue par Bouchard [27]. Une contribution de cette partie est d'abord d'augmenter la taille du jeu de données utilisé pour l'entraînement pour faciliter l'application de la méthode. Puis, une seconde contribution est d'analyser l'impact individuel de chacune des mesures statistiques utilisées par Bouchard afin de déterminer lesquels sont les plus importantes. Il en ressort que d'utiliser simplement la valeur maximale des données de la fenêtre temporelle permet une grande amélioration de la précision de la méthode, la portant à près de 95% des positions correctement identifiées.

Lors de la conception d'IPADL, seul le réseau de neurones de base de l'outil Weka a été considéré. Le domaine de l'apprentissage profond ayant connu d'importants développements depuis, il serait intéressant de vérifier si IPADL ne pourrait pas être amélioré en remplaçant sa forêt aléatoire par un réseau de neurones profond.

Le deuxième chapitre de la partie apporte la notion de série temporelle. De nombreux filtres existent dans la littérature pour les séries temporelles. Ces filtres pourraient possiblement se révéler plus efficaces que les filtres simples actuellement inclus dans IPADL. Un filtre d'intérêt est le filtre de Kalman [154], largement utilisé dans le domaine des séries temporelles en lien avec les méthodes de positionnement et de traçabilité.

IPADL n'a été utilisé que dans un contexte intérieur de maison intelligente. Il n'y a toutefois aucune raison de croire qu'il ne pourrait pas être utilisé ailleurs. Des travaux futurs pourraient se concentrer sur l'utilisation d'IPADL dans un contexte intérieur de grande surface, comme un entrepôt. Il peut aussi être intéressant de l'utiliser en plein air pour mieux comparer la méthode avec la technologie du GPS.

Retour sur les objectifs

À la suite de la revue de littérature sur les protocoles de communication et sur le positionnement intérieur, il est apparu qu'il y avait un besoin pour un protocole de communication différent faisant de la position des choses son élément central. Cinq objectifs ont été formulés à partir de ce besoin :

1. Concevoir un protocole de communication basé sur les positions des objets;
2. Concevoir un protocole de communication capable de gérer une durée de vie des messages;
3. Concevoir un système de positionnement intérieur rapide pour objets;
4. Identifier et vérifier les limitations du protocole conçu;
5. Déployer le protocole dans un scénario se rapprochant des conditions réelles.

Conclusion

Tous ces objectifs ont été atteints, en partie ou en totalité. L'existence même de cette thèse témoigne de l'atteinte des objectifs 1 et 2. Ce protocole se nomme le BMP. Comme il utilise la position des objets pour fonctionner et qu'obtenir cette position à l'intérieur présente un défi, cette thèse introduit la méthode IPADL pour obtenir cette position en temps réel [15] dans un environnement équipé d'antennes RFID. L'objectif 4 n'est pas totalement atteint. Certaines limites ont été identifiées, comme le nombre d'objets et la fréquence des messages. Toutefois, elles n'ont pas pu être mesurées avec précision pour établir un guide des conditions d'utilisation du BMP. L'atteinte partielle de cet objectif n'a cependant pas empêché l'achèvement de l'objectif 5. Le BMP a été déployé avec succès dans le DOMUS dans un scénario rappelant la séquence d'activités d'une personne rentrant chez elle tard le soir.

Perspective

L'élaboration d'un protocole de communication est une tâche complexe faisant intervenir un grand nombre de domaines de l'informatique. Ce protocole étant orienté vers l'IdO, des compétences en électronique auront aussi été nécessaires. La notion de position demande aussi l'apport des mathématiques et de la physique. Le BMP ne pourrait que gagner à être amélioré par des experts de ces différents domaines. Dans le même ordre d'idées, les applications souhaitant utiliser un BMP devraient inclure des spécialistes du domaine d'application dans leur équipe de développement. Particulièrement, l'application de l'errance nocturne doit impérativement inclure des ergothérapeutes et des médecins dans l'équipe pour s'assurer de ne pas nuire à la personne assistée. De même, l'intersection intelligente ne devrait pas être développée sans l'apport d'ingénieurs civils et d'urbanistes. La science s'oriente vers la multidisciplinarité et l'informatique doit servir de point d'union fort.

BIBLIOGRAPHIE

- [1] M. Altini, D. Brunelli, E. Farella, et L. Benini, « Bluetooth indoor localization with multiple neural networks, » dans *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on*. IEEE, 2010, pp. 295–300.
- [2] M. Azizyan, I. Constandache, et R. Roy Choudhury, « SurroundSense : Mobile phone localization via ambience fingerprinting, » dans *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*. ACM, 2009, pp. 261–272.
- [3] A. Ayimdji Tekemetieu, C. Haidon, F. Bergeron, H. Kenfack Ngankam, H. Pigot, C. Gouin-Vallerand, et S. Giroux, « Context modelling in ambient assisted living : Trends and lessons, » dans *Internet of Things Cases and Studies*, F. Garca Marquez et B. Lev, éditeurs. Springer, 2021.
- [4] L. Atzori, A. Iera, et G. Morabito, « The internet of things : A survey, » *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [5] L. Atzori, A. Iera, et G. Morabito, « Understanding the Internet of Things : Definition, potentials, and societal role of a fast evolving paradigm, » *Ad Hoc Networks*, vol. 56, pp. 122–140, mar 2017.
- [6] D. W. Aha, D. Kibler, et M. K. Albert, « Instance-based learning algorithms, » *Machine Learning*, vol. 6, no. 1, pp. 37–66, 1991.
- [7] Z. Alliance, « ZigBee specification FAQ, » ZigBee Alliance. Rapport technique, 2013, [Online ; accessed 27-April-2018]. Disponible à <https://web.archive.org/web/20130627172453/http://www.zigbee.org/Specifications/ZigBee/FAQ.aspx>
- [8] L. Alliance, « LoRa specification, » Lora Alliance, Rapport technique, 2015.
- [9] R. Arora, « Comparative analysis of classification algorithms on different datasets using WEKA, » *International Journal of Computer Applications*, vol. 54, no. 13, 2012.

Bibliographie

- [10] K. Ashton, « That “internet of things” thing, » *RFID journal*, pp. 97–117, 2009.
- [11] R. Al-Shaqi, M. Mourshed, et Y. Rezgui, « Progress in ambient assisted systems for independent living by the elderly, » *SpringerPlus*, vol. 5, no. 1, p. 624, 2016.
- [12] K. Bouchard, B. Bouchard, et A. Bouzouane, « Guidelines to efficient smart home design for rapid AI prototyping : a case study, » dans *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2012, p. 29.
- [13] K. Bouchard, B. Bouchard, et A. Bouzouane, « Spatial recognition of activities for cognitive assistance : Realistic scenarios using clinical data from Alzheimer’s patients, » *Journal of Ambient Intelligence and Humanized Computing*, vol. 5, no. 5, pp. 759–774, 2014.
- [14] F. Bergeron, K. Bouchard, S. Gaboury, S. Giroux, et B. Bouchard, « Indoor positioning system for smart homes based on decision trees and passive RFID, » dans *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2016, pp. 42–53.
- [15] F. Bergeron, K. Bouchard, S. Giroux, S. Gaboury, et B. Bouchard, « Real-time constraints for activities of daily living recognition, » dans *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2016, pp. 1–2.
- [16] F. Bergeron, K. Bouchard, S. Gaboury, et S. Giroux, « Tracking objects within a smart home, » *Expert Systems With Applications*, vol. 113, pp. 428–442, 2018.
- [17] F. Bergeron, K. Bouchard, S. Giroux, et S. Gaboury, « RFID indoor localization using statistical features, » *Cybernetics and Systems*, 2021.
- [18] F. Baccelli, B. Blaszczyzyn, et P. Muhlethaler, « An Aloha protocol for multihop mobile wireless networks, » *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 421–436, feb 2006.
- [19] J. Brusey, C. Floerkemeier, M. Harrison, et M. Fletcher, « Reasoning about uncertainty in location identification with RFID, » dans *Workshop on Reasoning with Uncertainty in Robotics at IJCAI*, 2003, pp. 23–30.

- [20] K. Bouchard, D. Fortin-Simard, S. Gaboury, B. Bouchard, et A. Bouzouane, « Accurate trilateration for passive RFID localization in smart homes, » *International Journal of Wireless Information Networks*, vol. 21, no. 1, pp. 32–47, 2013.
- [21] L. Breiman, J. Friedman, C. J. Stone, et R. A. Olshen, *Classification and Regression Trees*. CRC press, 1984.
- [22] C. Belley, S. Gaboury, B. Bouchard, et A. Bouzouane, « An efficient and inexpensive method for activity recognition within a smart home based on load signatures of appliances, » *Pervasive and Mobile Computing*, vol. 12, pp. 58–78, 2014.
- [23] F. Bergeron, S. Giroux, K. Bouchard, et S. Gaboury, « RFID based activities of daily living recognition, » dans *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*. IEEE, 2017, pp. 1–5.
- [24] R. Barco, P. Lazaro, et P. Munoz, « A unified framework for selfhealing in wireless networks, » *IEEE Communications Magazine*, vol. 50, no. 12, pp. 134–142, dec 2012.
- [25] R. R. Bouckaert, « Bayesian network classifiers in weka for version 3-5-7, » *Artificial Intelligence Tools*, vol. 11, no. 3, pp. 369–387, 2008.
- [26] K. Bouchard, « Unsupervised spatial data mining for human activity recognition based on objects movement and emergent behaviors, » Thèse de doctorat, Université du Québec à Chicoutimi, 2014.
- [27] K. Bouchard, « Statistical features for objects localization with passive RFID in smart homes, » dans *International Conference on Smart Objects and Technologies for Social Good*. Springer, 2017, pp. 21–30.
- [28] L. Breiman, « Random forests, » *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

Bibliographie

- [29] K. Bouchard, R. Ramezani, et A. Naeim, « Features based proximity localization with Bluetooth emitters, » dans *Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual. IEEE*, 2016, pp. 1–5.
- [30] A. Bekkali, H. Sanson, et M. Matsumoto, « RFID indoor positioning based on probabilistic RFID map and Kalman filtering, » dans *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on. IEEE*, 2007, pp. 21–21.
- [31] J. Burroughs, « X-10QR home automation using the PIC16F877A, » *Lamp*, vol. 10, no. 10, 2010.
- [32] Q. Chen, S. Abdelwahed, et A. Erradi, « A model-based validated autonomic approach to self-protect computing systems, » *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 446–460, oct 2014.
- [33] M. Chan, E. Campo, D. Estève, et J.-Y. Fourniols, « Smart homes—Current features and future perspectives, » *Maturitas*, vol. 64, no. 2, pp. 90–97, 2009.
- [34] D. J. Cook, A. S. Crandall, B. L. Thomas, et N. C. Krishnan, « CASAS : A smart home in a box, » *Computer*, vol. 46, no. 7, pp. 62–69, 2013.
- [35] K. Christidis et M. Devetsikiotis, « Blockchains and smart contracts for the internet of things, » *Ieee Access*, vol. 4, pp. 2292–2303, 2016.
- [36] L. Calderoni, M. Ferrara, A. Franco, et D. Maio, « Indoor localization in a hospital environment using random forest classifiers, » *Expert Systems with Applications*, vol. 42, no. 1, pp. 125–134, 2015.
- [37] G. F. Cooper et E. Herskovits, « A Bayesian method for the induction of probabilistic networks from data, » *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992.
- [38] V. Chawla et D. S. Ha, « An overview of passive RFID, » *IEEE Communications Magazine*, vol. 45, no. 9, pp. 11–17, sep 2007.
- [39] S. S. Chawathe, « Beacon placement for indoor localization using bluetooth, » dans *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on. IEEE*, 2008, pp. 980–985.

- [40] D. J. Cook et N. Krishnan, « Mining the home environment, » *Journal of Intelligent Information Systems*, vol. 43, no. 3, pp. 503–519, 2014.
- [41] C.-Y. Cheng, T.-H. Lan, et C.-H. Chan, « An improved localization algorithm with wireless heartbeat monitoring system for patient safety in psychiatric wards, » *Engineering Applications of Artificial Intelligence*, vol. 26, no. 2, pp. 905–912, 2013.
- [42] V. Coskun, B. Ozdenizci, et K. Ok, « A survey on near field communication (NFC) technology, » *Wireless Personal Communications*, vol. 71, no. 3, pp. 2259–2294, aug 2013.
- [43] L. Corporation, « IEEE 802.11ac : What does it mean for test ? » LitePoint Corporation, Rapport technique, 2013, [Online ; accessed 27-April-2018]. Disponible à http://litepoint.com/whitepaper/80211ac_Whitepaper.pdf
- [44] P. Chahuara, F. Portet, et M. Vacher, « Making context aware decision from uncertain information in a smart home : A markov logic network approach, » dans *International Joint Conference on Ambient Intelligence*. Springer, 2013, pp. 78–93.
- [45] Y. Chen, W. Shen, H. Huo, et Y. Xu, « A smart gateway for health care system using wireless sensor network, » dans *2010 Fourth International Conference on Sensor Technologies and Applications*. IEEE, jul 2010, pp. 545–550.
- [46] B. P. Crow, I. Widjaja, L. Kim, et P. T. Sakai, « IEEE 802.11 wireless local area networks, » *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116–126, 1997.
- [47] J. Chai, C. Wu, C. Zhao, H.-L. Chi, X. Wang, B. W.-K. Ling, et K. L. Teo, « Reference tag supported RFID tracking using robust support vector regression and Kalman filter, » *Advanced Engineering Informatics*, vol. 32, pp. 1–10, 2017.
- [48] L. De Nardis et M.-G. Di Benedetto, « Medium access control design for UWB communication systems : Review and trends, » *Journal of Communications and Networks*, vol. 5, no. 4, pp. 386–393, dec 2003.

Bibliographie

- [49] M. Dopico, A. Gomez, D. De la Fuente, N. García, R. Rosillo, et J. Puche, « A vision of industry 4.0 from an artificial intelligence point of view, » *Proceedings of the 2016 International Conference on Artificial Intelligence, ICAI 2016 WORLDCOMP 2016*, pp. 407–413, 2016.
- [50] E. De Poorter, I. Moerman, et P. Demeester, « Enabling direct connectivity between heterogeneous objects in the internet of things through a network-service-oriented architecture, » *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 1, p. 61, dec 2011.
- [51] A. O. de Sá, N. Nedjah, et L. de Macedo Mourelle, « Distributed efficient localization in swarm robotics using Min–Max and particle swarm optimization, » *Expert Systems with Applications*, vol. 50, pp. 55–65, 2016.
- [52] F. Despau, A. van den Bossche, K. Jaffrès-Runser, et T. Val, « NTWR : An accurate time-of-flight-based N-ary ranging protocol for Ultra-Wide band, » *Ad Hoc Networks*, vol. 79, pp. 1–19, oct 2018.
- [53] S. Diagne, T. Val, A. K. Farota, B. Diop, O. Assogba *et al.*, « Performances analysis of a system of localization by angle of arrival UWB radio, » *International Journal of Communications, Network and System Sciences*, vol. 13, no. 02, p. 15, 2020.
- [54] C. Dobre et F. Xhafa, « Intelligent services for big data science, » *Future Generation Computer Systems*, vol. 37, pp. 267–281, jul 2014.
- [55] Y. Dai, Y. Xiang, et G. Zhang, « Self-healing and hybrid diagnosis in cloud computing, » dans *IEEE International Conference on Cloud Computing*. Springer, 2009, pp. 45–56.
- [56] F. Eibe, M. A. Hall, I. H. Witten, et J. Pal, « The WEKA workbench, » *Online Appendix for “Data Mining : Practical Machine Learning Tools and Techniques*, vol. 4, 2016.
- [57] A. Fernández-Ares, A. M. Mora, M. G. Arenas, P. García-Sánchez, G. Romero, V. Rivas, P. A. Castillo, et J. J. Merelo, « Studying real traffic and mobility scenarios for a Smart City using a new monitoring and tracking system, » *Future Generation Computer Systems*, vol. 76, pp. 163–179, nov 2017.

- [58] M. Dopico, A. Gomez, D. De la Fuente, N. García, R. Rosillo, et J. Puche, « A vision of industry 4.0 from an artificial intelligence point of view, » *Proceedings of the 2016 International Conference on Artificial Intelligence, ICAI 2016 WORLDCOMP 2016*, pp. 407–413, 2016.
- [59] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat, et O. K. Tonguz, « Self-organized traffic control, » dans *Proceedings of the seventh ACM international workshop on Vehicular InterNetworking VANET '10*. New York, New York, USA : ACM Press, 2010, p. 85.
- [60] M. Fahim, I. Fatima, S. Lee, et Y.-K. Lee, « Daily life activity tracking application for smart homes using android smartphone, » dans *Advanced Communication Technology (ICACT), 2012 14th International Conference on*. IEEE, 2012, pp. 241–245.
- [61] B. Fouladi et S. Ghanoun, « Security evaluation of the Z-Wave wireless protocol, » *Black Hat USA*, vol. 24, pp. 1–2, 2013.
- [62] E. Fadel, V. Gungor, L. Nassef, N. Akkari, M. A. Malik, S. Almasri, et I. F. Akyildiz, « A survey on wireless sensor networks for smart grid, » *Computer Communications*, vol. 71, pp. 22–33, nov 2015.
- [63] J. Friedman, T. Hastie, R. Tibshirani *et al.*, « Additive logistic regression : A statistical view of boosting (with discussion and a rejoinder by the authors), » *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.
- [64] P. Fraga-Lamas, T. M. Fernández-Caramés, Ó. Blanco-Novoa, et M. A. Vilar-Montesinos, « A review on industrial augmented reality systems for the industry 4.0 shipyard, » *IEEE Access*, vol. 6, pp. 13 358–13 375, 2018.
- [65] U. A. Force, « GPS Accuracy, » U.S. Air Force, Rapport technique, 2017, [Online ; accessed 27-April-2018]. Disponible à <https://www.gps.gov/systems/gps/performance/accuracy/>
- [66] D. Fortin-Simard, J.-S. Bilodeau, K. Bouchard, S. Gaboury, B. Bouchard, et A. Bouzouane, « Exploiting passive RFID technology for activity recognition in smart homes, » *IEEE Intelligent Systems*, vol. 30, no. 4, pp. 7–15, 2015.

Bibliographie

- [67] D. Fortin-Simard, K. Bouchard, S. Gaboury, B. Bouchard, et A. Bouzouane, « Accurate passive RFID localization system for smart homes, » dans *Networked Embedded Systems for Every Application (NESEA), 2012 IEEE 3rd International Conference on. IEEE*, 2012, pp. 1–8.
- [68] J. Gama, « Functional trees, » *Machine Learning*, vol. 55, no. 3, pp. 219–250, 2004.
- [69] S. Giroux, T. Leblanc, A. Bouzouane, B. Bouchard, H. Pigot, et J. Bauchet, « The Praxis of Cognitive Assistance in Smart Homes. » dans *BMI Book*, 2009, pp. 183–211.
- [70] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, et B. Heile, « IEEE 802.15.4 : A developing standard for low-power low-cost wireless personal area networks, » pp. 12–19, sep 2001.
- [71] J.-S. Gutmann et C. Schlegel, « Amos : Comparison of scan matching approaches for self-localization in indoor environments, » dans *Advanced Mobile Robot, 1996., Proceedings of the First Euromicro Workshop on. IEEE*, 1996, pp. 61–67.
- [72] J. Hightower et G. Borriello, « Location systems for ubiquitous computing, » *Computer*, vol. 34, no. 8, pp. 57–66, 2001.
- [73] Y.-L. Hsu, P.-H. Chou, H.-C. Chang, S.-L. Lin, S.-C. Yang, H.-Y. Su, C.-C. Chang, Y.-S. Cheng, et Y.-C. Kuo, « Design and implementation of a smart home system using multisensor data fusion technology, » *Sensors*, vol. 17, no. 7, p. 1631, 2017.
- [74] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, et I. H. Witten, « The WEKA data mining software : An update, » *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
- [75] J. He, Y. Geng, et K. Pahlavan, « Toward accurate human tracking : Modeling time-of-arrival for wireless wearable sensors in multipath environment, » *IEEE Sensors Journal*, vol. 14, no. 11, pp. 3996–4006, 2014.
- [76] A. Harter, A. Hopper, P. Steggle, A. Ward, et P. Webster, « The anatomy of a context-aware application, » *Wireless Networks*, vol. 8, no. 2/3, pp. 187–197, 2002.

- [77] D. Hedley et M. Jacobs, « The shape of things to come : The Equifax breach, the GDPR and open-source security, » *Computer Fraud & Security*, vol. 2017, no. 11, pp. 5–7, nov 2017.
- [78] T. hoon Kim, C. Ramos, et S. Mohammed, « Smart City and IoT, » pp. 159–162, nov 2017.
- [79] D. P. Hutabarat, D. Patria, S. Budijono, et R. Saleh, « Human tracking application in a certain closed area using RFID sensors and IP camera, » dans *Information Technology, Computer, and Electrical Engineering (ICITACEE), 2016 3rd International Conference on*. IEEE, 2016, pp. 11–16.
- [80] G. Holmes, B. Pfahringer, R. Kirkby, E. Frank, et M. Hall, « Multiclass alternating decision trees, » dans *Machine Learning: ECML 2002*. Springer, 2002, pp. 161–172.
- [81] T. Hargreaves et C. Wilson, *Smart Homes and their Users*. Springer, 2017.
- [82] G. P. Joshi, S. Acharya, C.-S. Kim, B.-S. Kim, et S. W. Kim, « Smart solutions in elderly care facilities with RFID system and its integration with wireless sensor networks, » *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.
- [83] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, et C. Shahabi, « Big data and its technical challenges, » *Communications of the ACM*, vol. 57, no. 7, pp. 86–94, jul 2014.
- [84] J. Jin, J. Gubbi, T. Luo, et M. Palaniswami, « Network architecture and QoS issues in the internet of things for a smart city, » dans *2012 International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, oct 2012, pp. 956–961.
- [85] D. A. Jumeily, A. Hussain, et P. Fergus, « Using adaptive neural networks to provide self-healing autonomic software, » *International Journal of Space-Based and Situated Computing*, vol. 5, no. 3, p. 129, 2015.
- [86] M. Z. Khan, B. Askwith, F. Bouhafs, et M. Asim, « Limitations of simulation tools for large-scale wireless sensor networks, » dans *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*. IEEE, 2011, pp. 820–825.

Bibliographie

- [87] J. O. Kephart et D. M. Chess, « The vision of autonomic computing, » *Computer*, vol. 36, no. 1, pp. 41–50, 2003.
- [88] N. C. Krishnan et D. J. Cook, « Activity recognition on streaming sensor data, » *Pervasive and Mobile Computing*, vol. 10, pp. 138–154, 2014.
- [89] S. Kuila, N. Dhanda, S. Joardar, et S. Neogy, « Analytical survey on standards of Internet of things framework and platforms, » dans *Emerging Technologies in Data Mining and Information Security*. Springer, 2019, pp. 33–44.
- [90] S.-C. Kim, Y.-S. Jeong, et S.-O. Park, « RFID-based indoor location tracking to ensure the safety of the elderly in smart home environments, » *Personal and Ubiquitous Computing*, vol. 17, no. 8, pp. 1699–1707, 2013.
- [91] G. Kortuem, F. Kawsar, V. Sundramoorthy, et D. Fitton, « Smart objects as building blocks for the Internet of things, » *IEEE Internet Computing*, vol. 14, no. 1, pp. 44–51, jan 2010.
- [92] N. Kwok, D. Liu, et G. Dissanayake, « Evolutionary computing based mobile robot localization, » *Engineering Applications of Artificial Intelligence*, vol. 19, no. 8, pp. 857 – 868, 2006.
- [93] J. Knighten, S. McMillan, T. Chambers, et J. Payton, « Recognizing social gestures with a wrist-worn smartband, » dans *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*. IEEE, 2015, pp. 544–549.
- [94] H. Kenfack Ngankam, H. Pigot, D. Lorrain, I. Viens, et S. Giroux, « Context awareness architecture for ambient-assisted living applications : Case study of nighttime wandering, » *Journal of Rehabilitation and Assistive Technologies Engineering*, vol. 7, p. 2055668319887864, 2020.
- [95] R. Kohavi, « Scaling up the accuracy of naive-bayes classifiers : A decision-tree hybrid, » dans *KDD*. Citeseer, 1996, pp. 202–207.
- [96] J. Kurose et K. Ross, « Computer networking : A top down approach, 2012. »

- [97] M. Kourogi, N. Sakata, T. Okuma, et T. Kurata, « Indoor/outdoor pedestrian navigation with an embedded GPS/RFID/self-contained sensor system, » *Advances in Artificial Reality and Tele-Existence*, pp. 1310–1321, 2006.
- [98] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, et M. Hoffmann, « Industry 4.0, » *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, 2014.
- [99] N. Landwehr, M. Hall, et E. Frank, « Logistic model trees, » *Machine Learning*, vol. 59, no. 1-2, pp. 161–205, 2005.
- [100] J. Lessmann, P. Janacik, L. Lachev, et D. Orfanus, « Comparative study of wireless network simulators, » dans *Seventh International Conference on Networking (icn 2008)*. IEEE, 2008, pp. 517–523.
- [101] X. Liu, J. Peng, et T. Liu, « A novel indoor localization system based on passive RFID technology, » dans *Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on*. IEEE, 2011, pp. 4285–4288.
- [102] Y. Lu, « Industry 4.0 : A survey on technologies, applications and open research issues, » *Journal of Industrial Information Integration*, vol. 6, pp. 1–10, jun 2017.
- [103] W. Lumpkins, « Home automation : Insteon (X10 Meets Powerline) [product reviews], » *IEEE Consumer Electronics Magazine*, vol. 4, no. 4, pp. 140–144, oct 2015.
- [104] I. Majeed et O. Arif, « Non-linear eigenspace visual object tracking, » *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 363 – 374, 2016.
- [105] E. Morganti, L. Angelini, A. Adami, D. Lalanne, L. Lorenzelli, et E. Mugellini, « A smart watch with embedded sensors to recognize objects, grasps and forearm gestures, » *Procedia Engineering*, vol. 41, pp. 1169–1175, 2012.
- [106] Martin Woolley, « Bluetooth 5. Go Faster. Go Further. » Bluetooth Special Interest Group, Rapport technique, 2018, [Online ; accessed 27-April-2018]. Disponible à <https://www.bluetooth.com/bluetooth-technology/bluetooth5/bluetooth5-paper>

Bibliographie

- [107] P. Misra et P. Enge, *Global Positioning System : Signals, Measurements and Performance*, Second Edition. Lincoln, MA : GangaJamuna Press, 2006.
- [108] P. Meng, K. Fehre, A. Rappelsberger, et K.-P. Adlassnig, « Framework for near-field-communication-based geo-localization and personalization for Android-based smartphones—Application in hospital environments, » *Stud Health Technol Inform*, vol. 198, pp. 9–16, 2014.
- [109] E. Manavalan et K. Jayakrishna, « A review of Internet of Things (IoT) embedded sustainable supply chain for industry 4.0 requirements, » *Computers & Industrial Engineering*, vol. 127, pp. 925– 953, jan 2019.
- [110] C. Morato, K. N. Kaipa, B. Zhao, et S. K. Gupta, « Toward safe human robot collaboration by using multiple kinects based real-time human tracking, » *Journal of Computing and Information Science in Engineering*, vol. 14, no. 1, p. 011006, 2014.
- [111] H. D. Mehr, H. Polat, et A. Cetin, « Resident activity recognition in smart homes by using artificial neural networks, » dans *Smart Grid Congress and Fair (ICSG), 2016 4th International Istanbul*. IEEE, 2016, pp. 1–5.
- [112] G. Mulligan, « The 6LoWPAN architecture, » dans *Proceedings of the 4th Workshop on Embedded Networked Sensors, EmNets 2007*. New York, New York, USA : ACM Press, 2007, pp. 78–82.
- [113] H. Ma, Y. Wang, et K. Wang, « Automatic detection of false positive RFID readings using machine learning algorithms, » *Expert Systems with Applications*, vol. 91, pp. 442–451, 2018.
- [114] L. M. Ni, Y. Liu, Y. C. Lau, et A. P. Patil, « LANDMARC : Indoor location sensing using active RFID, » *Wireless Networks*, vol. 10, no. 6, pp. 701–710, 2004.
- [115] W. Ni, W. Wu, et K. Li, « A message efficient intersection control algorithm for intelligent transportation in smart cities, » *Future Generation Computer Systems*, vol. 76, pp. 339–349, nov 2017.

- [116] C.-W. Ou, C.-J. Chao, F.-S. Chang, S.-M. Wang, G.-X. Liu, M.- R. Wu, K.-Y. Cho, L.-T. Hwang, et Y.-Y. Huan, « A ZigBee position technique for indoor localization based on proximity learning, » dans *Mechatronics and Automation (ICMA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 875–880.
- [117] C. H. S. Oliveira, S. Giroux, H. Ngankam, et H. Pigot, « Generating bayesian network structures for self-diagnosis of sensor networks in the context of ambient assisted living for aging well, » dans *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10461 LNCS. Springer, Cham, 2017, pp. 198–210.
- [118] I. Oppermann, M. Hämäläinen, et J. Iinatti, *UWB : Theory and Applications*. John Wiley & Sons, 2005.
- [119] W. H. Organization, *World Report on Ageing and Health*. World Health Organization, 2015.
- [120] J. Ocepek, A. E. K. Roberts, et G. Vidmar, « Evaluation of treatment in the smart home IRIS in terms of functional independence and occupational performance and satisfaction, » *Computational and Mathematical Methods in Medicine*, vol. 2013, pp. 1–10, nov 2013.
- [121] J. Postel *et al.*, « User datagram protocol, » 1980.
- [122] B. W. Parkinson, P. Enge, P. Axelrad, et J. J. Spilker Jr, *Global Positioning System : Theory and Applications, Volume II*. American Institute of Aeronautics and Astronautics, 1996.
- [123] H. Pigot et S. Giroux, « Living labs for designing assistive technologies, » dans *E-health Networking, Application & Services (HealthCom), 2015 17th International Conference on*. IEEE, 2015, pp. 170–176.
- [124] K. Pahlavan, P. Krishnamurthy, et Y. Geng, « Localization challenges for the emergence of the smart world, » *IEEE Access*, vol. 3, pp. 3058–3067, 2015.
- [125] J. Postel, « RFC 0791 : Internet protocol, » 1981. Disponible à <https://www.hjp.at/doc/rfc/rfc791.html>

Bibliographie

- [126] J. Postel, « Rfc0793 : Transmission control protocol, » 1981.
- [127] H. Pirsiavash et D. Ramanan, « Detecting activities of daily living in first-person camera views, » dans *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2847–2854.
- [128] J. R. Quinlan, *CA. 5 : Programs for Machine Learning*. Elsevier, 2014.
- [129] L. Qiao, Z. Zheng, W. Cui, et L. Wang, « A survey on Wi-Fi HaLow technology for Internet of things, » dans *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*. IEEE, oct 2018, pp. 1–5.
- [130] S. Rampfl, « Network simulation and its limitations, » dans *Proceeding zum seminar future internet (FI), Innovative Internet Technologien und Mobilkommunikation (ITM) und autonomous communication networks (ACN)*, vol. 57. Citeseer, 2013.
- [131] M. Resnik, « Turtles termites and traffic jams, » 1994.
- [132] J. Rossey, I. Moerman, P. Demeester, et J. Hoebeke, « Wi-Fi helping out Bluetooth smart for an improved home automation user experience, » dans *Communications and Vehicular Technologies (SCVT), 2016 Symposium on*. IEEE, 2016, pp. 1–6.
- [133] R. Radziszewski, H. K. Ngankam, V. Grégoire, D. Lorrain, H. Pigot, et S. Giroux, « Designing calm and non-intrusive ambient assisted living system for monitoring nighttime wanderings, » *International Journal of Pervasive Computing and Communications*, vol. 13, no. 2, pp. 114–129, 2017.
- [134] R. Radziszewski, H. Ngankam, H. Pigot, V. Grégoire, D. Lorrain, et S. Giroux, « An Ambient Assisted living nighttime wandering system for elderly, » dans *18th International Conference on Information Integration and Web based Application & Services (IIWAS)*, Singapore, 2016.
- [135] R. S. Ransing et M. Rajput, « Smart home for elderly care, based on wireless sensor network, » dans *Nascent Technologies in the Engineering Field (ICNTE), 2015 International Conference on*. IEEE, 2015, pp. 1–5.
- [136] M. Satyanarayanan, « The Emergence of Edge Computing, » *Computer*, vol. 50, no. 1, pp. 30–39, jan 2017.

- [137] S. Singh et I. Chana, « EARTH : Energy-aware autonomic resource scheduling in cloud computing, » *Journal of Intelligent & Fuzzy Systems*, vol. 30, no. 3, pp. 1581–1600, mar 2016.
- [138] S. Singh et I. Chana, « QoS-aware autonomic cloud computing for ICT, » dans *Proceedings of International Conference on ICT for Sustainable Development*. Springer, 2016, pp. 569–577.
- [139] W. Shi et S. Dustdar, « The Promise of Edge Computing, » *Computer*, vol. 49, no. 5, pp. 78–81, may 2016.
- [140] K. P. Subbu, B. Gozick, et R. Dantu, « LocateMe : Magnetic-fieldsbased indoor localization using smartphones, » *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 4, p. 73, 2013.
- [141] S. Seol, E. K. Lee, et W. Kim, « Indoor mobile object tracking using RFID, » *Future Generation Computer Systems*, vol. 76, pp. 443–451, nov 2017.
- [142] K. Scarfone et J. Padgette, « Guide to bluetooth security, » *NIST Special Publication*, vol. 800, no. 2008, p. 121, 2008.
- [143] N. A. Somani et Y. Patel, « Zigbee : A low power wireless technology for industrial applications, » *International Journal of Control Theory and Computer Modelling (IJCTCM)*, vol. 2, no. 3, pp. 27–33, 2012.
- [144] E. Stip et V. Rialle, « Environmental cognitive remediation in schizophrenia : ethical implications of “smart home” technology. » *Canadian Journal of Psychiatry. Revue canadienne de psychiatrie*, vol. 50, no. 5, pp. 281–91, apr 2005.
- [145] N. B. Suryavanshi, K. Viswavardhan Reddy, et V. R. Chandrika, « Direction finding capability in Bluetooth 5.1 standard, » dans *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, LNICST*, vol. 276. Springer, Cham, feb 2019, pp. 53–65.
- [146] A. A. Tole *et al.*, « Big data challenges, » *Database Systems Journal*, vol. 4, no. 3, pp. 31–40, 2013.

Bibliographie

- [147] O. Tekdas et V. Isler, « Sensor placement for triangulation-based localization, » *IEEE transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 681–685, 2010.
- [148] R. Tesoriero, R. Tebar, J. A. Gallud, M. D. Lozano, et V. M. R. Penichet, « Improving location awareness in indoor spaces using RFID technology, » *Expert Systems with Applications*, vol. 37, no. 1, pp. 894–898, 2010.
- [149] N. Uribe-Pérez et C. Pous, « A novel communication system approach for a Smart City based on the human nervous system, » *Future Generation Computer Systems*, vol. 76, pp. 314–328, nov 2017.
- [150] I. Unwala, Z. Taqvi, et J. Lu, « Thread : An IoT protocol, » dans *2018 IEEE Green Technologies Conference (GreenTech)*. IEEE, apr 2018, pp. 161–167.
- [151] J. Vanus, M. Cerny, et J. Koziorek, « The proposal of the smart home care solution with KNX components, » dans *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, jul 2015, pp. 1–5.
- [152] A. Van Den Bossche, R. Dalcé, N. Gonzalez, et T. Val, « LocURa : A new localisation and UWB-based ranging testbed for the Internet of things, » dans *IPIN 2018 9th International Conference on Indoor Positioning and Indoor Navigation*. IEEE, sep 2018, pp. 1–6.
- [153] A. Van Den Bossche, R. Dalce, et T. Val, « OpenWiNo : An open hardware and software framework for fast-prototyping in the IoT, » dans *2016 23rd International Conference on Telecommunications, ICT 2016*. IEEE, may 2016, pp. 1–6.
- [154] G. Welch, G. Bishop *et al.*, « An introduction to the Kalman filter, » 1995.
- [155] D. Wu et C. Butz, « On the complexity of probabilistic inference in singly connected bayesian networks, » dans *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*. Springer, 2005, pp. 581–590.
- [156] G. I. Webb, « Decision tree grafting from the all-tests-but-one partition, » dans *IJCAI*. Citeseer, 1999, pp. 702–707.

- [157] I. H. Witten, E. Frank, M. A. Hall, et C. J. Pal, *Data Mining : Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2016.
- [158] O. J. Woodman et R. K. Harle, « Concurrent scheduling in the Active Bat location system, » dans *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, mar 2010, pp. 431–437.
- [159] Wikipedia contributors, « Radio-frequency identification — Wikipedia, The Free Encyclopedia, » 2018, [Online ; accessed 27-April-2018]. Disponible à https://en.wikipedia.org/w/index.php?title=Radio-frequency_identification&oldid=836773617
- [160] D. Washburn, U. Sindhu, S. Balaouras, R. A. Dines, N. Hayes, et L. E. Nelson, « Helping CIOs understand “smart city” initiatives, » *Growth*, vol. 17, no. 2, pp. 1–17, 2009.
- [161] J. Yim, « Introducing a decision tree-based indoor positioning technique, » *Expert Systems with Applications*, vol. 34, no. 2, pp. 1296–1302, 2008.
- [162] E. Ziouva et T. Antonakopoulos, « CSMA/CA performance under high traffic conditions : throughput and delay analysis, » *Computer Communications*, vol. 25, no. 3, pp. 313–321, feb 2002.
- [163] A. Zanella, N. Bui, A. Castellani, L. Vangelista, et M. Zorzi, « Internet of things for smart cities, » *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, feb 2014.
- [164] A. L. Zulas, A. S. Crandall, et M. Schmitter-Edgecombe, « Caregiver needs from elder care assistive smart homes children of elder adults assessment, » dans *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 58, no. 1. SAGE Publications, 2014, pp. 634–638.
- [165] M. Zareei, A. Islam, S. Baharun, C. Vargas-Rosales, L. Azpilicueta, et N. Mansoor, « Medium access control protocols for cognitive radio ad hoc networks : A survey, » *Sensors*, vol. 17, no. 9, p. 2136, sep 2017.

Bibliographie

- [166] H. Zimmermann, « OSI reference model-the ISO model of architecture for open systems interconnection, » *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425–432, 1980.
- [167] J. Zheng et M. J. Lee, « A comprehensive performance study of IEEE 802.15. 4, » *Sensor Network Operations*, vol. 4, pp. 218–237, 2006.

DÉPÔT LÉGAL : MAI 2024

Frédéric Bergeron est détenteur d'un baccalauréat et d'un doctorat en informatique de l'Université de Sherbrooke. Il s'est également mérité en 2017 un prix Octas, dans la catégorie relève étudiante, remis par le réseau Action TI, en récompense de l'un de ses projets de fin de baccalauréat.

À la recherche d'une science concrète et appliquée, Frédéric Bergeron quitte Thetford et s'inscrit à l'Université de Sherbrooke, d'abord en physique, puis l'an suivant en informatique. Un travail d'assistant de recherche en domotique appliquée à une clientèle en perte d'autonomie l'attire en recherche en 2015. Après quelques mois, son travail devient une maîtrise, puis une thèse de doctorat, terminée en février 2021. Enseignant au Cégep de Thetford depuis 2022, Frédéric occupe ses temps libres à rêver de jardinage, d'urbanisme et autres optimisations environnementales.



Ce livre s'intéresse aux communications entre les objets intelligents alors que les méthodes actuelles de fonctionnement ont été établies à l'époque de la communication entre postes informatiques fixes.

L'ouvrage propose un nouveau protocole de communication abandonnant ces idées pour plutôt s'appuyer sur la notion de position des objets. Le protocole se nomme BMP, pour *Bounded Message Protocol*. Ce protocole a pour caractéristiques principales l'envoi de messages en mode diffusion et l'arrêt de la propagation à une certaine distance du point d'émission. Les messages de ce protocole présentent aussi une durée de vie au bout de laquelle chaque message doit être détruit. Ce protocole est conçu pour être léger avec un en-tête minimal à son fonctionnement. Deux expériences sont réalisées. Une première permet de vérifier le bon fonctionnement des mécanismes de contrôle du BMP. Une deuxième propose un scénario imitant la vie réelle dans un appartement intelligent et confirme que le BMP fonctionne dans ce type d'environnement.

Parallèlement au *Bounded Message Protocol*, l'ouvrage présente l'IPADL (pour *Indoor Positioning for Activities of Daily Living*), une méthode de positionnement d'objets devant permettre l'usage du BMP à l'intérieur des bâtiments, là où connaître sa position précise est plus complexe qu'utiliser un système satellitaire.