



HAL
open science

Thing-to-thing context-awareness at the edge

Mariem Harmassi

► **To cite this version:**

Mariem Harmassi. Thing-to-thing context-awareness at the edge. Other [cs.OH]. Université de La Rochelle, 2019. English. NNT : 2019LAROS037 . tel-03646659

HAL Id: tel-03646659

<https://theses.hal.science/tel-03646659v1>

Submitted on 19 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



LA ROCHELLE
UNIVERSITÉ



ECOLE DOCTORALE (EUCLIDE)

[Laboratoire Informatique, Image et Interaction (L3i)]

Thèse de Doctorat

Discipline: Informatique et Applications

THÈSE présentée par :

Mariem Harmassi

Thing-to-Thing context-awareness at the edge

Soutenance 03 Septembre 2019 devant le jury composé de :

M. Emmanuel Chaput	Professeur des Universités, ENSEEIHT - INP Toulouse, Rapporteur
Mme Nathalie Mitton	Directrice de Recherche, INRIA Lille - Nord Europe, Rapporteur
Mme Hakima Chaouchi	Professeur, Institut Mines Telecom - Telecom Sud Paris, Examineur
Mme Françoise Sailhan	Maître de conférences, CNAM Paris, Examineur
M. Yacine Ghamri-Doudane	Professeur des Universités, La Rochelle Université, Directeur de thèse
M. Cyril Faucher	Maître de conférences, La Rochelle Université, Co-encadrant

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Prof. Yacine Ghamri-Doudane and Cyril Faucher for their continuous support of my Ph.D study and related research, for the patience, motivation, and immense knowledge they provide me. The flexibility and freedom they gave me to propose new ideas in line with the strategies and objectives of the thesis. I wish to thank them for giving me the chance to widen my research from various perspectives and helping me build my momentum as a researcher.

Besides my advisor, I would like to thank the rest of my thesis committee : Prof. Emmanuel Chaput, Prof. Nathalie Mitton, Prof. Hakima Chaouchi, and Dr. Françoise Sailhan, for accepting to examine this dissertation. Special thanks goes also to Prof. Emmanuel Chaput and Prof. Nathalie Mitton, for generously offering their time and good will to serve as my thesis reviewers).

I thank my fellow lab-mates for the rich and useful discussions and the good moment we spent. Also I thank my friends for their moral support during stressful and difficult moments. My sincere thanks also goes to Prof. Jean-Marc Ogier, President of the University of La Rochelle, Karim Aït-Mokhtar, Director of Doctoral school, Yacine Ghamri-Doudane the Director of the L3I Lab as well as all the Doctoral school and L3I Lab Staff who contributed every day to made it even easier.

Last but not the least, I would like to thank my family, specially my parents, Mr. Abdelhak Harmassi and Mrs. Kalthoum Harmassi, for all the support and encouragements they gave me all along my life and especially throughout this doctoral degree .

Abstract

Internet of Things (IoT) today comprises a plethora of different sensors and diverse connected objects, constantly collecting and sharing heterogeneous sensory data from their environment. This enables the emergence of new applications exploiting the collected data towards facilitating citizens lifestyle. These IoT applications are made context-aware thanks to data collected about user's context, to adapt their behavior autonomously without human intervention. In this Thesis, we propose a novel paradigm that concern Machine to Machine (M2M)/Thing To Thing (T2T) interactions to be aware of each other context named "**T2T context-awareness at the edge**", it brings conventional context-awareness from the application front end to the application back-end. More precisely, we propose to empower IoT devices with intelligence, allowing them to understand their environment and adapt their behaviors based on, and even act upon, the information captured by the neighboring devices around, thus creating a collective intelligence.

The first challenge we face in order to make IoT devices context-aware is (i) How can we extract such information without deploying any dedicated resources for this task? To do so we propose in our first work a **context reasoner** [1] based a cooperation among IoT devices located in the same surrounding. Such cooperation aims at mutually exchange data about each other context.

To enable IoT devices to see, hear, and smell the physical world for themselves, we need firstly to make them connected to share their observations. For a mobile and energy-constrained device, the second challenge we face is (ii) How to discover as much neighbors as possible in its vicinity while preserving its energy resource? We propose Welcome [2] a **Low latency and Energy efficient neighbor discovery scheme** that is based on a single-delegate election method.

Finally, a Publish-Subscribe that takes into account the context at the edge of IoT devices, can greatly reduce the overhead and save the energy by avoiding unnecessary transmission of data that doesn't match application requirements. However, if not thought about properly building such T2T context-awareness could imply an overload of subscriptions to meet

context-estimation needs. So our third contribution is (iii) How to make IoT devices context-aware while saving energy. To answer this, We propose an **Energy efficient and context-aware Publish-Subscribe** [3] that strike a balance between energy-consumption due to context estimation and energy-saving due to context-based filtering near to data sources.

Keywords : Internet of Things (IoT), IoT Applications, Services, Quality of context, Quality of service ,Context-awareness, Machine learning, Conditional Random Field, Neighbor Discovery, Publish-Subscribe, Mobility, Energy Efficiency, low Latency, Wireless sensor networks (WSN), Game Theory, Dynamic Coalitional Games.

Résumé

L'Internet des objets (IdO) comprend aujourd'hui une riche offre d'objets connectés, qui permettent de collecter et de partager en continu des données hétérogènes se rapportant à leurs environnements. Ceci a permis l'émergence d'un nouveau type d'applications, qui sont basées sur ces données et permettent de faciliter la vie des citoyens. Ces applications de l'Internet des objets sont dites « sensibles au contexte ». Grâce aux données collectées sur le contexte de l'utilisateur, elles sont en mesure d'adapter leur comportement de manière autonome, sans intervention humaine. Dans cette thèse, nous proposons un nouveau paradigme autour des interactions objet-à-objet, nommé « Interactions objet-à-objet pour la sensibilité au contexte en bordure de réseaux ». Ce dernier, permet de tenir compte d'un nouveau type de contexte, paradoxalement à la notion conventionnelle de « sensibilité au contexte » qui se limite au contexte de l'utilisateur d'une application. Ainsi nous proposons de nous intéresser pour la première fois au contexte des objets en tant que composante même de l'application. Cette thèse vise à doter les objets connectés d'un certain degré d'intelligence, leur permettant de comprendre leur propre environnement et d'en tenir compte dans leurs interactions objet-à-objet. Les contributions majeures de cette thèse se focalisent sur deux modules principaux. Nous proposons, dans un premier temps, un module d'identification de contextes capable de capter les contextes des objets mobiles et de délivrer ce genre d'information de contexte de façon exacte et continue. Sur la base de cette information de contexte assurée par le premier module, nous proposons un deuxième module de collecte de données sensible aux contextes de déploiement des objets connectés.

Afin que ceci soit possible, de nombreux verrous restent à lever. Concernant le premier module d'identification de contexte, le premier défi rencontré afin de permettre aux objets connectés de devenir sensibles au contexte est (i) Comment peut-on assurer une identification de contexte exacte pour des objets déployés dans des environnements incontrôlables ? Pour ce faire, nous proposons dans notre premier travail un raisonneur dédié à l'apprentissage et le raisonnement sur le contexte [1]. Le raisonneur proposé est fondé sur une stratégie coopérative entre les différents dispositifs IdO d'un même voisinage. Cette coopération vise à un échange mutuel des données parmi les ressources disponibles d'un même voisinage. La deuxième problématique rencontrée est (ii) Comment peut-on assurer une identification de contexte continue pour des nœuds mobiles appartenant à des réseaux opportunistes ?

Nous devons tout d'abord leur permettre de découvrir un maximum de voisins afin d'établir un échange avec. Afin de répondre à cette deuxième problématique nous proposons WELCOME un protocole de découverte des voisinages éco énergétique et à faible latence [2] qui permettra de diminuer considérablement les collisions sur la base d'une découverte de voisinage à faible coût en termes de latence et d'énergie.

La troisième problématique, se rapportant au module de collecte de données sensible au contexte, est (iii) Comment peut-on assurer une collecte efficace et précise sur la base du contexte physique de déploiement des capteurs. En effet, d'une part tenir compte de l'information de contexte des capteurs, permet d'éviter toutes transmissions inutiles ou redondante de données. D'autre part, la contextualisation des données implique un partage et donc des transmissions de messages. La question ici (iii) Comment peut-on contextualiser au mieux le plus grand nombre d'objets connectés tout en préservant au mieux leurs ressources énergétiques. Afin de répondre à cette question, nous proposons un Publish-Subscribe à la fois sensible au contexte et éco énergétique basé sur un jeu coalitionnel dynamique qui permet de résoudre ces conflits d'intérêts entre les sources dans un réseaux [3]. **Mots clés :** Internet des Objets (IdO), IdO Applications, Services, Qualité de contexte, Qualité de service, Sensibilisation au Contexte, Apprentissage Machine, Les Champs Aléatoires conditionnels, Découverte de voisinage, Publish-Subscribe, mobilité, Efficacité énergétique, Faible latence, Réseau de capteurs, Théorie de jeux, Jeux Coalitionnels Dynamiques.

Table des matières

1	Introduction	1
1.1	Motivation	1
1.2	Contributions	3
1.3	Thesis Organization	5
2	Context-awareness in Internet of Things - State of the Art	8
2.1	Introduction	8
2.2	Context-awareness	2
2.3	Context-aware sensor selection	4
2.4	Neighbor Discovery in Duty-cycled wireless Sensor Network	7
2.5	Context-aware Publish-Subscribe systems	2
2.6	Discussion	4
2.7	Conclusion	6
3	Smart things : T2T Context-awareness at the edge Reasoner	9
3.1	Introduction	9
3.2	Background : Conditional Random Field	2
3.3	CRF for Context Awareness at the Edge	3
3.3.1	System Model	4
3.3.1.1	Network spatio-temporal parameters	4
3.3.1.2	CRF key parameters	4
3.3.2	Find Co-existing Objects Influence Relationship	5
3.3.2.1	Temporal Association Potential	5
3.3.2.2	Spatial Association Potential	8
3.3.2.3	Spatio-temporal Interaction Potential	1
3.3.2.4	Spatio-temporal CRF	2
3.4	Performance Evaluation	1
3.4.1	Simulation over a synthetic dataset	1
3.4.1.1	Potential Functions	2
3.4.1.2	Synthetic data set	4
3.4.1.3	Simulation scenario based a synthetic dataset	4
3.4.1.4	Testing : Impact of increasing sensors (Simulations)	5
3.4.2	Simulation over a real dataset	9
3.5	Conclusions	10
4	WELCOME, Low Latency and Energy Efficient Neighbor Discovery for Mobile and IoT Devices	14

4.1	Introduction	49
4.2	Overview of Neighbor Discovery Schemes	50
4.3	WELCOME: Towards a Flock Discovery	53
4.3.1	System Model	53
4.3.2	Delegate Node Eligibility	54
4.3.3	Node activity schedule in WELCOME	55
4.3.3.1	Delegate Node	56
4.3.3.2	Member Node	57
4.3.4	Absence of WELCOME message - Self-organization	58
4.4	Analytical Evaluation	60
4.4.1	Evaluation Metrics	60
4.5	Theoretical Comparison	61
4.5.1	key parameter	61
4.5.2	Detailed theoretical analysis	62
4.5.2.1	Discussion	65
4.6	Simulation-based Evaluation	66
4.6.1	Simulation Scenario	66
4.6.2	Simulation Results	68
4.6.2.1	Average discovered neighbors	68
4.6.2.2	Latency vs Discovery	71
4.6.2.3	Energy vs Discovery	71
4.7	Conclusions	73
5	Context-aware and Energy Efficient Publish-Subscribe for IOT: CEEPS4IOT	76
5.1	Introduction	76
5.2	CEEPS4 IoT	78
5.2.1	System Model	78
5.2.2	Deriving Object Utility	79
5.2.2.1	Gain function	80
5.2.2.2	Cost function	80
5.2.2.3	Utility function	81
5.2.3	Coalitional Game formation	82
5.2.3.1	Collaborative Context at the Edge Algorithm	82
5.2.3.2	Coalition Stability and Convergence	83
5.3	Performance Evaluation	86
5.3.1	NS3 Simulation-based evaluation	87
5.3.1.1	Implementation	87
5.3.1.2	Evaluations scenarios	88
5.3.2	Evaluation Results	89
5.3.2.1	Throughput	89
5.3.2.2	Latency	90
5.3.2.3	Energy	91
5.4	Conclusions	92
6	Conclusion and Future works	95
6.1	Contributions	95
6.1.1	Context Reasoner	96

6.1.2	WELCOME	96
6.1.3	CEEPS4IoT	97
6.2	Future Works	98

List of Figures

1.1	Context-awareness at the edge example	2
2.1	Comparison of neighbor discovery schemes	17
3.1	IoT spatio-temporal chain CRF Model vs spatio-temporal chain CRF model	28
3.2	Temporal associations	31
3.3	Spatial association	33
3.4	Spatio-temporal interaction	34
3.5	Spatio-temporal IoT CRF model	36
3.6	Varying nodes vs distance and time	41
3.7	Effect of age of data variations on context estimation (Test-bed with 5 SensorTags)	44
4.1	Discovery initialization - no delegate node detected	55
4.2	Discovery with existing delegate node message detection	55
4.3	Average discovered neighbors comparison	68
4.4	Histogram of average discovered neighbors	69
4.5	CDF comparison of average discovered neighbors	69
4.6	Latency vs discovery	72
4.7	Energy vs discovery	72
5.1	Partition function Γ	83
5.2	Coalition formation through Merge and split operations	84
5.3	Throughput	89
5.4	Latency	90
5.5	Energy	91

Chapter 1

Introduction

Contents

1.1 Motivation	1
1.2 Contributions	3
1.3 Thesis Organization	5

1.1 Motivation

The rapid worldwide deployment of Internet passed from connecting HTML pages to interconnecting all physical and digital objects, things, creating a new ecosystem called “Internet of Things” (IoT). IoT enables the emergence of a new generation of applications and services destined to improve our quality of life, security, and performance. These IoT applications always rely on a number of “smart things” to help resolve the challenges (e.g. road traffic congestion) we face every day. A smart thing is an identifiable physical element that is potentially able to communicate, act, sense, and interact with the environment and other smart things. The realization of Smart Cities, which is a major consumer of IoT applications, requires a huge number of connected sensors, or actuators as instances of “smart things”, also referred as IoT devices/objects or simply things, to act autonomously to collect data and to provide different services to IoT application consumers.

Given the frequent mobility of these things and their dynamic states change (e.g. sleep/wake up, connect/disconnect), IoT network is a highly unstructured cloud. The environment in which these devices are deployed can largely impact the collected data and the provided

services quality. So, in this thesis we advocate that it is important that such devices are made aware of the context of their surroundings, particularly while collecting data. Thus the provision of IoT applications that could satisfy the user needs is still an open issue. To supply IoT systems with data collected by mobile devices, we need a better understanding of how the data was collected and communicated to other machines.

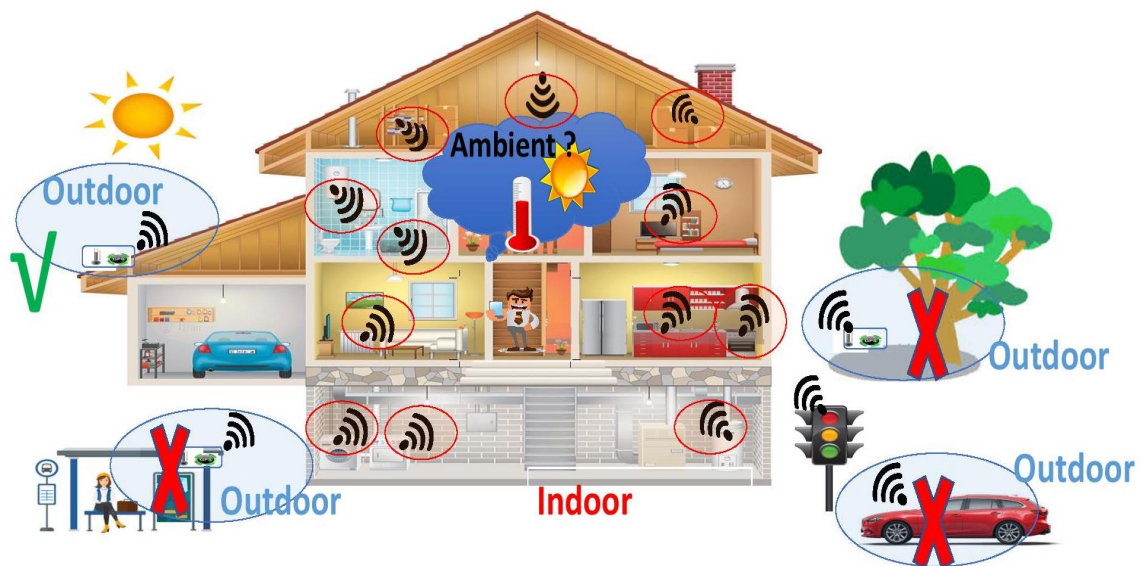


Figure 1.1: Context-awareness at the edge example

To better assess our motivation, let us present the example in 1.1, where an IoT application user, want to know the environment quality outside his home. To answer this request the IoT application needs to be provided with some sensing data such as the ambient temperature, the humidity, the carbon dioxide (CO₂), the volatile organic compounds (VOCs), outside his home. Hence, the IoT application will firstly search for all available necessary sensors lying into the desired area. Let us consider the specific case of temperature sensor selection only. The list of available candidates to answer the application request comprises indoor and outdoor sensors. As it is clear that the indoor temperature sensors doesn't meet the application requirements, they should be removed from the temperature sensor list. From those deployed outdoor it seems that not all of them can satisfy the application requirement. Indeed, their situational contexts influence their reports. For instance the sensor installed inside the car will report ambient temperature inside the car, and the sensor under the tree will report outdoor under shade temperature. It seems that the temperature sensor on the roof is the locally best candidate to answer the application

request. This is the one that needs to be identified and selected. On the light of described scenario, “*context-aware computing at the edge*” avoid IoT applications to be provided with undesired data with regard to IoT application requirements. More generally, as things provide their functions as real-world services, we consider essential to fill that gap by proposing a new form of context-awareness that provide IoT applications with context-aware services.

On the one hand, this may offer innovative opportunities for IoT applications/users to specify precisely their needs (temperature under shade, ambient luminosity, etc). On the other hand, this allow things to participate and act autonomously based on their context. More precisely, we focus on machine to machine communication and data exchange to ensure providing IoT system with trustworthy and fully described data. Our intent, in this thesis, is to empower IoT devices with a layer of intelligence for the development of a **T2T consciousness/environmental awareness**. To mimic humans talk, we aim to boost T2T communication to a much higher level than simple communication, we want to contextualize these communication to make it a conversation. To do so, we tackle in this thesis the main IoT modules that could ensure T2T context-awareness at the edge:

- Context identification module: (i) How to ensure a trustworthy and accurate context estimation for the IoT devices especially when deployed in uncontrolled environments?
(ii) How to ensure the continuity of such an context identification module for mobile nodes in opportunistic networks?
- Context-aware data collect module: How to provide applications with the appropriate data in efficient and?

1.2 Contributions

The first question we faced in order to make IoT devices context-aware is: how to annotate raw data gathered by connected objects according to their context of acquisition (i) without deploying any dedicated resources and (ii) at the same time ensure a trustworthy and accurate context recognition? We firstly propose a collaborative approach between connected objects in the same geographical location (neighborhood) to contextualize the data collected during the collection phase. We think that any pair of sensors reporting information about their environment to the applications back-end, can at the same time

report information about their mutual environment. For instance, a temperature sensor near a luminosity sensor can report its reading to describe its neighbor’s context, and vice-versa. Based on this, we propose a “**T2T context Reasoner**” which, once the sensor reading stream and the observations about its physical context reported by its neighbors is provided, returns an estimation about sensor’s context at a given instant as well as its temporal evolution. By doing so, the reasoner assigns a sensing context to each sensor reading. The dedicated reasoner is based on Conditional Random Fields (CRF), a well known statistical model method used for structured prediction. CRF allows to take into account the spatio-temporal interaction between sensors of the same neighborhood in order to decode a temporal sequence. Results show that the proposed model estimates the context of a sensor with high accuracy (up to **98.5%**). The proposed model not only estimates the context of a sensor with high accuracy but also strongly overcome biased data.

To enable IoT devices to see, hear, and smell the physical world for themselves, we need firstly to make them connected to share their observations. Mobile and energy-constrained devices in opportunistic networks face the challenge of: How to discover as much neighbors as possible in its vicinity while preserving energy resources? Indeed, T2T context-awareness at the edge is based on a collective intelligence among IoT devices in the same neighborhood, which requires wireless low-power devices in each other proximity to interact locally to form an opportunistic network. To achieve this, IoT devices need to discover themselves first. As IoT devices are energy-constrained and opting for duty cycling, neighbor discovery under such constrains is a major challenge. To address this question, we propose as a second contribution, “**Welcome**”: A low-latency and energy-efficient neighbor discovery scheme that copes with heterogeneity of duty-cycles and avoids collision to occur based on a single-delegate election algorithm. The single-delegate election algorithm is based a self-organization and rotation of the delegate role among spatio-temporally co-located nodes. In this way only one node at time can broadcast messages as a delegate node for a neighborhood during a period of time.

We perform simulation over NS3 for a 100-node neighborhood size, opting for 5% duty cycle. Welcome is compared to indirect schemes *i)* Blend [4] and *ii)* Garissidi [5], *iii)* fixed-slot based Hello [6], and *iv)* G-Nihao [7], *v)* dynamic slot based Searchlight [8] and *vi)* Blinddate [9], and *vii)* stochastic Birthday mechanism [10]. Results show that Welcome

not only discovered 100% neighbors but also yielded low energy consumption and discovery latency due to the reduction in the amount of transmitted messages leading to collisions.

The third and last challenge concerns context-aware Publish-Subscribe protocol. Our aim here is to answer the question of: How to optimally manage the data transmissions based on source's context? Once a Publish-Subscribe system is aware of the desired context by an IoT application on the one hand and the measured data context in the other hand, a targeted selection from available resources is possible and only the ones that match the IoT application requirements are involved. However, this selection phase requires a context estimation process that could entail further subscriptions, which in turn results in additional messages transmissions. Indeed, messages transmissions required by context estimation phase depends on the number of participants and their frequency of participation. There is a trade-off between two conflicting objectives here: Maximizing the number of participants in the context estimation process and minimizing the overall energy consumption. To solve this optimization problem, we propose, "**CEEPS4IoT**" as a context-aware Publish-Subscribe based on a dynamical coalitional game between co-located IoT devices. **CEEPS4IoT** incentivizes the best set of neighbors to share data for the context estimation task while dealing with IoT devices individual rationality to save their resources.

We evaluate our proposed "CEEPS4IoT" compared to a Publish-Subscribe with a location-based filtering feature, using NS3 simulations. For scalability matter, we consider 50, 100, 200 and 300 nodes, and for each network size we set the half of nodes as publishers and the other half as subscribers for two types of data (temperature and luminosity). Results show that CEEPS4IoT, is an energy efficient context-aware publish-subscribe system. Our proposed model could not only reduce energy consumption by 50% up to 75%, but also reduce the network latency by 30% up to 70% compared to the selected Publish-Subscribe model. In addition, CEEPS4IoT can reach almost 50% of throughput compared to only 30% for the compared model.

1.3 Thesis Organization

The remaining of the document is organized as follows. Chapter 2 highlights the background along a review of the state of the art context-awareness, neighbor discovery and data collection schemes in the literature. The proposed "T2T context-awareness at the edge

reasoner' is presented in Chapter 3 as the core engine of a novel smart data management system. Chapter 4 ensures the continuity of the context identification task for mobile nodes deployed in opportunistic networks, based on WELCOME a neighbor discovery protocol. In Chapter 5, we present and validate a context-aware and energy-efficient Publish-subscribe protocol. The Chapter 6 concludes our work with a discussion on future insight.

Chapter 2

Context-awareness in Internet of Things - State of the Art

Contents

2.1 Introduction	8
2.2 Context-awareness	10
2.3 Context-aware sensor selection	11
2.4 Neighbor Discovery in Duty-cycled wireless Sensor Network	14
2.5 Context-aware Publish-Subscribe systems	18
2.6 Discussion	20
2.7 Conclusion	22

2.1 Introduction

We are entering a new era of computing technology, the era of Internet of Things (IoT). Internet of Things (IoT) today comprises sensors, actuators and other connected devices providing data to IoT applications. Massively deployed IoT devices enable IoT applications to adapt their behavior to the user's environment (e.g. location, activity, nearby people). This later notion, known as context awareness, allows IoT applications to support an improved response ability and thus it plays an important role in the IoT. However, that large amount of IoT devices, context-aware applications need to rely on, are characterized by their mobility and/or the dynamic nature of their deployment. Hence, they can change

their position and/or environment at any time. Although the fact that popularization of data collection is a real opportunity to take advantage of the widely deployed sensors, it is also a big challenge. Indeed, most of those sensors will be deployed by different users in different environments, and for different purposes. This massive deployment and data collected from IoT devices results, at the same time, in a better spatio-temporal coverage of urban area, creating new interesting opportunities, but also implies several challenges.

For instance, a temperature sensor is designed to sense the temperature, though, the context in which it is deployed (indoor/outdoors, in ambient/in shade, on a human body) results in different readings. These are influenced by the physical conditions of the device's surroundings. So, the belief motivating this thesis is that we need a thorough investigation of IoT application's services context. We propose in this thesis a new paradigm, that we name "**T2T context-awareness at the edge**", which allow IoT devices to be aware of their own context as well as the one of their counterparts IoT devices during an interaction phase when triggered by an application. In order to better understand the present notion of context-awareness and its major challenges, in the following sections we investigate this through the following main questions:

- What is context-awareness? Does it exist definitions and concept related to IoT device's context ?
- If Yes, which context features are mainly considered by these approaches to select the appropriate sensors to answer an application request?
- Selection is first of all based on a discovery process and especially for mobile nodes deployed in opportunistic, thus we investigate how such a fundamental IoT application needs was addressed in the literature: Does neighbor discovery mechanisms allow nodes in each other communication range to discover all their neighbors while running on low duty cycles?
- Publish-Subscribe systems that encompass context-aware sensor selection schemes are named context-aware Publish-Subscribe: How such architectures claim to reduce energy-consumption and overhead while enhance the throughput?

To answer these questions, we firstly review the existing definitions of context-awareness and investigate the relevant works on context-aware sensor selection, in sections ??, [Section 2.2](#) and [Section 2.3](#), respectively. Next, in [Section 2.4](#), we discuss an overview of existing

neighbor discovery schemes in the literature and identify their drawbacks, we also provide a preliminary simulation-based comparison among the best schemes, depicting the need for a new scheme. In [Section 2.5](#), we present different Context-aware Publish-Subscribe systems concentrating on their energy consumption challenges and context-awareness concerns. Accordingly, in [Section 2.6](#), we briefly expose our upcoming proposed contributions as an answer to the identified issues T2T context-aware computing.

2.2 Context-awareness

Context-aware computing is first introduced by Schilit and Theimer in 1994 [11]. Thereafter, it gained notoriety with the emergence of “ubiquitous computing” and “ambient intelligence” (AmI). Context-aware computing under the paradigm of AmI is limited to ambient living environments where applications have to be accommodated to some environment events in a predefined way. Thanks to smart cities, that benefits from a great amount of connected objects, which is also keeping increasing every day, IoT brings context-aware computing to open scenarios. IoT enable a new generation of applications to emerge, named “context-aware applications”. Such applications are capable of acting autonomously on behalf of user’s behavior taking into account different context information (physical context, computational context, and user context). Context-aware applications have to successfully identify the environment conditions user is part of, and then react autonomously to the environment and influencing it by running the processes that trigger actions and create services with or without direct human intervention. Two types of context are considered in the literature: the context of users and the context of things.

Context of users From the literature, many researchers tried to introduce the concept of context-awareness. Existing works until 1999 tried to define it through examples,[11] [12] [13]. Schilit et al. (1994)[11] define context as location, identities of nearby people and objects. Brown (1997)[12] enumerates location, identities of the people around the user, the time of day, season, temperature. Rayn et al. (1997) [13] define it as user’s location, environment, identity and time. However such definitions are limited to the mentioned list of context entities, any new entity not cited in the definition become hardly considered as a context feature. In order to overcome such a drawback, the first general context definition was given by Dey and Abowd [14] in 1999. One of the most widely used definition: “*Context*

is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.” [14]. In 2005, Chaari [15] claimed that the latter do not separate application data from contextual information. Chaari proposed to define it as “all external parameters to the application that may affect its behavior by defining new views of its data and services”.

Context of things In 2012, He and coauthors [16] introduce the “context of things” concept. They propose to analyze the situational context of things in order to invoke appropriate services to react autonomously to the physical world events. They allow machines to understand things. They pointed out the lack of an explicit definition of the context of things and propose the following definition: ““Context of things” is a concept that enables people to understand things and translate their understanding into machine languages, in order to provide services for things automatically”.

In the above mentioned works, context-awareness aims to understand for a specific application, the users’ context, to offer them in an autonomous way the more appropriate services, where the user could be either a person [11] [12] [14] [15] or an object [16] [17]. What the above definitions miss is that context-awareness does not only concern the user application side. Context-awareness computing could be expanded to the application back-end (The whole IoT network and its associated applications). Indeed, Machine to Machine M2M interactions could be context aware enabling IoT devices to sense, perceive and understand their own environment and take decisions during their lifetime. Similarly, to context of user notion, IoT devices have to be aware of their own context and context of other objects they interact with, in order to adapt their behavior at run time without any human intervention.

2.3 Context-aware sensor selection

Due to the increasing number of available sensors, available connected devices can cover user requests with a better spatio-temporal granularity. However, they have the potential to send undesired data, altered data, or data with poor quality. To avoid these drawbacks, sensor selection is a key feature for an IoT application and many sensor selection algorithms

have been proposed. From the literature we classify sensor selection algorithms into three main classes. The first category uses keywords corresponding to some of the sensor context attributes (mainly location, time-stamp and sensor type) as selection criterias [18] [19] [20] [21]. The second category addresses the energy efficiency issue. In this one, the selection phase in addition consider measurement properties (e.g. accuracy, reliability, response time, frequency, measurement range), the operating properties (e.g. operating power range, system (sensor) lifetime), and survivability properties (e.g. cost of data transmission, cost of data generation) [22] [23] [24]. The third category addresses the data quality and trustworthiness challenge. The proposed approaches here focus mainly on massive data collection. These approaches are based on the calculation of a confidence interval around an estimated true value. If the sensor data lies between the bounds of this interval it is selected [25] [26] [27].

Keyword-based approaches In [18], the authors propose a distributed management system using a keyword-based interface for both human and machine clients. It selects the node whose location better matches the query destination. [19] proposes a context-aware architecture for autonomous semantic service selection in the health-care domain. The selection is based on the evaluation of the matching degree between identified user context and available services preconditions. The proposed matching is a hybrid matchmaking approach fusing purely logic and fuzzy modeling. uBox [20] is a distributed resource management architecture for interconnecting Web of Things "WoT" with a local platform and a global gateway. It provides decentralized infrastructure that integrates hierarchical localization based search. uBox performs searches based on location tags and object/sensor classes/-types (e.g. hierarchy local/class/actuator/light). In [21], the authors propose a distributed web service discovery architecture for Web-of-Things where every device is described by a RESTfull API. The authors present an ontology for describing not only the properties of web services but also the method invocation mechanisms. A device communication protocol is described by a RESTdesc, an RDFbased notation to describe the specific capabilities of a service, instead of its parameters and modalities [28]. Unlike previous works, they propose to consider dynamic and static service properties. The service selection is based on some contextual parameters such as sensor type, location, value type and format. The authors allow the automatic integration of services by enhancing ontology with gate description,

however the service properties are still limited.

Measurement, operating and survivability-based approaches CASSARAM [22], is a platform for context-aware sensor search and selection in IoT. The authors select sensors based on a desired user requirement. Users can prioritize certain sensor characteristics while searching for appropriate sensors. These characteristics can be classified based on measurement (e.g. accuracy, reliability, response time, frequency, sensitivity, measurement range), operation (e.g. operating power range, system or sensor lifetime), and survival (e.g. data transmission cost, data generation cost). Similarly, [23] is an SOA-based building automation system, it dynamically coordinates devices/services in accordance with a pre-defined plan associated with each context. Moreover, the architecture allows users to select sensors based on their residual energy, context properties (e.g. location, sensing type), in addition to preferences relations between some optional requirements (e.g. accuracy, latency, and sensing range) as a comparison matrix. In [24], the authors provide a large-scale registry based service discovery protocol suitable for energy constrained and sleeping nodes, which minimizes the control overhead and reduces energy consumption. The solution selects optimal services based on service popularity, the remaining battery, the host up time and its location.

FASEM [29] is a framework that provides automatic and dynamic services monitoring when an event occurs in an ambient environment which supports a selection method for Context Interpretation Networks (CIN). FASEM allows users to define event rules to specify their goals to achieve for each specific context. Then, the services are selected based on their prior functionality class and their dynamic quality of service. They propose a Global Directory of Services (GDS) to share discovered services in an area then the selection can be achieved using one of the two defined modes: active or passive. Available services are dynamically evaluated by a quality measure, DQoS, based on service availability, response Time and Probability of Response.

Confidence interval-based approaches The authors in [25] present a cloud-assisted platform considering the link quality and spatio-temporal correlation of data to minimize energy consumption of sensor selection. A heuristic-based greedy algorithm is proposed

to select an optimal set of active sensor nodes that maximizes the data utility as well as achieves energy load balancing. The work in [26] considers changes in the data quality from different sensors due to the dynamically changing environment. It allows an application to first learn the correlations between time series data from sensors without any prior knowledge. The learned correlations are then applied to find the useful sensor data and reconstruct the data with low quality. [27] notices the need for a filtering process to increase the data reliability. To do so, they propose to select sensors based on a correlation coefficient and some predefined validity rules. The correlation coefficient is given by Pearsons sample correlation coefficient between the reference data and the collected data from the sensors. The validity rules can be arbitrarily defined based on a empirical knowledge of the data behavior (e.g. Average temperature in La Rochelle in July is $68^{\circ}F$ ($20^{\circ}C$)).

From our point of view, traditional context-aware service discovery approaches are not suitable for service discovery in IoT, due to the differences between real-world services collecting data in controlled environments and uncontrolled environments. As things provide their functionalities as real-world services, under different environment conditions, the data provided could be altered or the service provided could change. Indeed, due to the high dynamic nature of urban environments and the mobility of things, the service provided by such things change constantly. In IoT, things are designed for a specific purpose however the provided service could change while acting in uncontrolled environment. For example, an IoT device (e.g. a temperature sensor) crosses different environments, despite it is designed to do the same task it can offer a large amount of delivered services (e.g. temperature data indoor/ outdoor/ambient/at shadow/body feel temperature). To provide the user/application with the right service, we need a better understanding of how the data is collected by the devices and communicated to other machines. We need *context-awareness at the edge* to describe and share dynamic service functions and select the appropriate ones regardless the user requirements, which avoid useless transmissions.

2.4 Neighbor Discovery in Duty-cycled wireless Sensor Network

Neighbor discovery schemes can be classified into two categories: *i*) direct, where nodes discover only the neighbors from which they directly receive a message [30] [31], and *ii*)

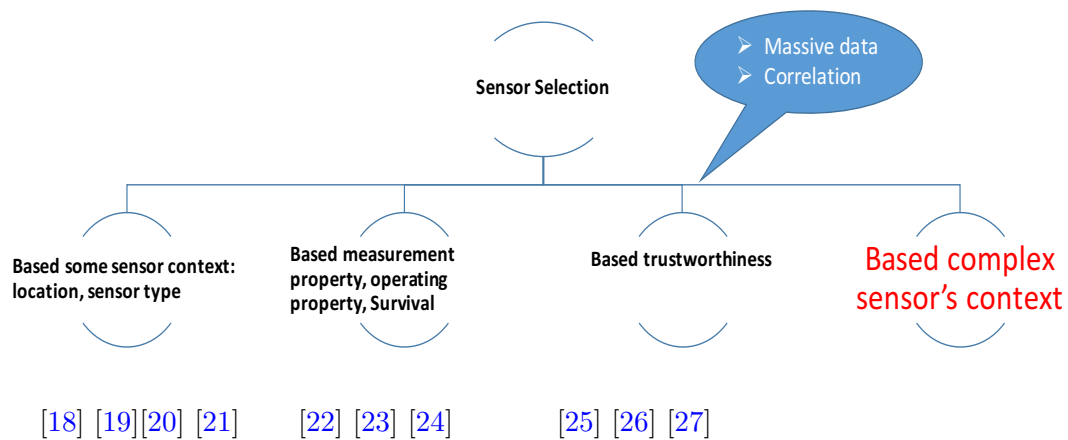


Table 2.1: Sensor selection protocols

indirect schemes, where nodes can learn the existence of neighbors indirectly from other nodes [32] [33] [34] [4] [5]. Direct schemes are further classified into *i*) quorum-based [35], *ii*) prime number-based [36] [37], *iii*) dynamic listen slot [9] [8], *iv*) fixed listen slot [6] [7], and *v*) stochastic [10]. These schemes can work on one or multiple frequency channels [38] [39] [40], and they all follow a similar principle of dividing time into slots and letting the node to be active in a slot based on a schedule defined by the respective algorithm.

Quorum-based schemes [35] guarantee that two nodes have at least one activity slot in common in a period of N slots by being active in \sqrt{N} slots. These mechanisms result in relatively high duty cycles and only function in homogeneous duty cycle conditions. The cyclic quorum design in heterogeneous duty cycle conditions is known as asymmetric design, and specific solutions are proposed to address this problem. Prime number-based asymmetric discovery schemes require a node to choose a single (e.g. U-Connect [37]) or a pair of prime numbers (e.g. Disco [36]) to derive its duty cycle. The activity slots of a node will be the multiples of the selected prime number(s). This approach can be extended, and differential codes can be built for each pair of nodes starting from relatively prime numbers [41]. Using results from number theory, it can be shown that any two nodes will finally wake-up on the same slot. The discovery latency in this case is the time slot corresponding to the product of the prime numbers used by the two nodes. The different

strategies also take different approaches in the activity slots. Disco proposes to send two beacons in each activity slot, one at the beginning and one at the end, and listen for incoming beacons from potential neighbors in the rest of the slot. The slot of U-Connect comprises a single beacon, followed by a listen period.

As the transmission and listen activities are independent and they can be conducted on different slots. In dynamic listen slot schemes, a large time period is divided into regular sized cycles, where each cycle is further composed of slots. Two types of slots exist, static transmission slots at fixed positions, either at the beginning or end of the cycle, and dynamic listen slots with a regular shift to either the left or right in consecutive cycles, up to the end of the period. Searchlight [8] is an example of such an approach, where a node has a static slot in the beginning of each cycle and an active slot shifted one slot to the right in each consecutive cycle. Similarly, BlindDate [9] uses one static slot in each cycle and two dynamic listen slots, one shifted to the right and one to the left in each consecutive cycle. A fixed schedule can also be used for listen slots. Nihao [7] takes the approach of *talk more listen less*, where more transmissions than listen slots exist in a given period. In the same context, Hello [6] is a highly parameterizable solution, where nodes listen more at the beginning of the period, and periodically wake up for transmissions. This scheme is shown to be a generalization of several other mechanisms, such as Disco, U-Connect and Searchlight. Finally, stochastic schemes such as Birthday [10] allow nodes to transmit beacons, listen for beacons from other nodes or sleep in a slot based on a probability distribution. Energy efficiency is ensured by choosing a lower probability for beacon transmission or for listening. Such schemes perform better on the average case compared with the deterministic approaches above, but they provide no bound on the worst case latency and they can lead to long tails in discovering the last fraction of nodes. The comparative analysis in [42] highlights the performance of direct schemes with respect to energy, latency and the neighbors discovered.

Indirect schemes include group-based discovery [34] which exploits existing schemes, such as Disco, and adapts a cooperative approach where nodes broadcast neighborhood table in an active slot. Similarly, Acc [32] improves the discovery phase in Disco by allowing nodes to share information regarding already discovered neighbors in their beacons, to the next encountering node. However, such exchange will result in high overhead, both for the individual node and the network, due to the large size of messages containing neighborhood

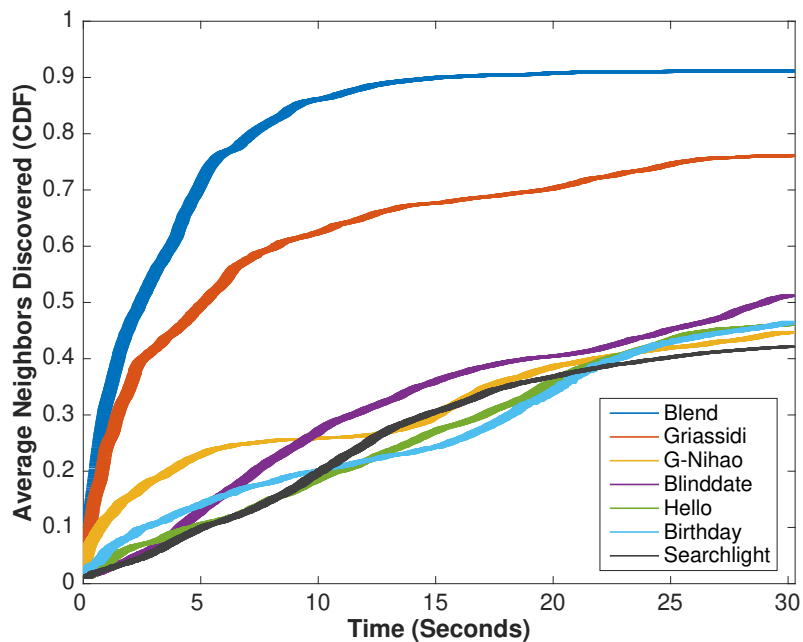


Figure 2.1: Comparison of neighbor discovery schemes

information, continuously exchanged between relatively low power nodes. EQS [33] is an extension to Quorum based system where nodes commonly active in a slot decide on a rendez-vous slot for their next wake up and, thereby, share the information regarding their respective discovered neighbor in the rendezvous slot. However, clock drift, as well as collisions, can lead to a node not waking up in the respective rendezvous slot. Further rendezvous-based indirect schemes are Blend [4] and Griassdi [5] where nodes share their next listen period in the transmitted message and any node receiving it wakes up at the rendezvous time to send additional beacon to ensure bi-directional discovery.

Figure 2.1 compares the best among the above mentioned schemes where is depicted the average neighbors discovered by each scheme with respect to time for a neighborhood size of 100 nodes. Clearly we observe that none of the existing schemes enable the nodes to discover all their neighbors. The recently proposed indirect schemes Blend and Griassdi perform the best, although they suffered from the long tail effect. For example, Blend discovered 90% of the neighbors in the first 10 seconds though exhibits a long tail afterward.

We believe, that multiple nodes transmitting beacons simultaneously, provoke collisions and prevent a fraction of nodes fail to discover their neighbors due to collisions. The

issue grows with the amount of nodes in the neighborhoods. Therefore, there is a need for a neighbor discovery mechanism where one node discovers and share the neighborhood information to the nodes in its communication range. Additionally, there is a need to reduce the number of messages exchanged over the wireless medium as well as the nodes energy consumption during a neighbor discovery process.

2.5 Context-aware Publish-Subscribe systems

Providing autonomous communication between intelligent devices is still a challenging task. Publish/Subscribe pattern constitutes a suitable solution for large-scale IoT deployments as it allows publishers and subscribers to be decoupled in time, space and synchronization. This decoupling enables the architecture to be highly scalable [43]. However Publish/Subscribe waste energy due to inefficient message routing [44]. Recently some works propose to make Publish/Subscribe systems context-aware through a dispatching strategy that limits the spreading of subscriptions only to those areas of the routing network where matching publishers context meets subscribers needs.

Antonic and coauthors [45] propose CloUd-based PUblish/Subscribe middleware (CUPUS) to process in near real-time big data streams. The proposed system is energy efficient as it avoids unnecessary transmission of sensor readings from Internet connected objects and filters out either redundant or irrelevant sensor readings. Indeed, the authors save energy by the means of (i) location-based subscriptions and announcements which reduce the number of subscriptions (ii) a matching of publications performed by mobile brokers near to the local publishers before they are sent on the cloud.

Lightweight XMPP [46] is a real-time Publish/Subscribe messaging systems based on IP technology and the extensible Markup Language (XML). In order to save energy, the publisher sends an adjusted data information according to the subscriber's needs, periodically, which avoids sending all the objects and attributes information and allows publishers to enter sleep mode. In [47], authors propose a middleware architecture to combine Software-Defined Network (SDN) with the Data Distribution Service (DDS) that provides real-time, scalable, and data-centric publish/subscribe capabilities. The solution reduces network traffic and energy consumption by the means of data filtering and data fusion mechanisms to drop unnecessary packets transmission and reduce number of forwarded packets. CoAP

[48] support both a request/response and resource-observe (a variant of Publish/Subscribe) application layer protocol, designed to meet constrained devices needs. It supports synchronous and asynchronous responses and runs over UDP to remove the TCP overhead and reduce bandwidth requirements. Thanks to the use of UDP as transport layer protocol and the header compression which reduces significantly the packet size, CoAP saves energy. MQTT [49] uses a hierarchical topic based Publish/Subscribe protocol suitable for battery-run devices. The Publish/Subscribe protocol frees the constrained devices from resource hungry operations like polling to frequently get the updated data. Thus, it decreases the network bandwidth and battery usage. Context-Aware System (CONASYS) [50], is a system able to provide context-aware information in order to contextualize entities and their data in IoT environments, all in real time. The main intention is to avoid the manual user intervention in the interpretation of the data and also facilitates the systems/entities interactions. Such an architecture decouples user request and responding devices. COLLECT [51] is a context-aware service oriented architecture based event-driven Publish/Subscribe. The main contribution of COLLECT, is that it facilitates context spreading and sharing among the nodes. The architecture is energy efficient through the use of an enterprise service bus 'ESB' makes the message broker light and scalable.

From the above mentioned literature, existing research works propose lightweight middleware systems. Some works tackle the problem through the use of ESB [51], or packet compression [48] [49]. Some other works choose content-based filtering and periodic publication as a solution [47] [46] [50]. On the other hand, some works propose to make Publish/Subscribe context-aware and prove that a context-based data filtering, while made close to data sources (mainly location, timestamp), leads to less energy consumption and decrease the network overhead as it reduces the number of exchanged messages [45]. However context considered in these works is limited to location and timestamp criteria. We suggest that expanding this context filter to some complex event could save more energy, reduce the overhead and at the same time ensures a better quality of service.

Scheme	Energy	Mobility	Architecture	Distibuted
CUPUS [45]	Yes	Yes	Event-driven Pub/Sub	Yes (Cloud/ mobile brokers)
Lightweight XMPP [46]	Yes	Yes	Periodical/ Event-driven Pub/Sub	Yes
MQTT [49]	Yes	N/A	Hierarchical topic Pub/Sub	No (Cloud)
CoAP [48]	Yes	N/A	REST Req/Res, Observer pattern	No
CONASYS [50]	Yes	N/A	Observer pattern	No
COLLECT [51]	Yes	Yes	Event-driven Publish/-Subscribe	Yes(fog/ Cloud)
[47]	Yes		content-based publish/-subscribe	Yes(virtualization)

Table 2.2: Summary of context-aware publish/subscribe systems

2.6 Discussion

The available definitions of context mainly consider the context of the user, where it can be either a person or an object. However, it is important to consider the physical context of things or sensors (i.e. at the service provider purpose, rather than at the consumer purpose) for an accurate data collection. For instance, to provide an accurate data for an application requesting ambient temperature, there is no need to involve temperature sensors under shade in the collect phase. Participatory sensing of undesired IoT objects could bias the IoT application in addition to the overhead and energy costs involved. However, the sensor selection criteria in most of the previously cited works are limited to the sensor’s location and its response time as context parameters. Thus, they ignore the characteristics and description of how and under which conditions the data are provided. Unlike existing works, we consider the IoT device’s environment to find its physical deployment context (e.g. underwater, outdoors or indoors, and thereby), going beyond typical characteristics defined in existing literature (type, location, time-stamp...). We propose to consider a complex notion of IoT device’s context, we want to bring IoT devices a new intelligence layer. Our objective is to make them aware of their own context at the edge (e.g. nearby sensors/devices, location, environment conditions, operating system, survival ...) and trigger some actions based on that. Hence, we propose a new paradigm, “T2T context-awareness at the edge”, that concerns IoT device’s situational context.

Our first work, highlights the IoT devices need to take into account their own context in their communication and decision tasks. An IoT device is deployed in an uncontrolled and dynamic environment. In addition, Its it can switches environments, randomly. That’s why, to learn IoT objects contexts is a difficult task, we need a tool for an accurate context estimation. We propose a context reasoner, as the “brain” component of an IoT middleware, capable of context sequence labeling. This main component is based on Conditional Random Field [52] which is fed with data streams upcoming from IoT devices in the same vicinity. CRF is a strong probabilistic model for structured prediction widely used in natural language processing, computer vision, and bio-informatics and many other domains. We use it to efficiently find the accurate spatio-temporal relations between different IoT devices in the same neighborhood and accurately estimate sequence contexts.

Energy efficient synchronous and asynchronous, symmetric and asymmetric duty cycled neighbor discovery schemes exists in the literature with a focus on minimizing the worst case latency required for the mutual discovery of a pair of low duty cycled devices. However, still a fraction of nodes in a neighborhood fail to find each other using existing neighbor discovery schemes. Such failures in discovery are mainly due to collisions between messages transmitted simultaneously by multiple nodes in the same slot where such phenomenon is never considered previously in the literature. To address this issue, we propose “Welcome”, a novel neighbor discovery scheme allowing a single node as a delegate, instead of all nodes in a proximity, to send beacons and listen to neighbors thereby reducing the overall amount of transmitted messages. This caters the issue of collisions between messages sent by concurrently transmitting nodes. We further allow nodes to rotate the role of becoming the delegate where each node compute its eligibility to become the delegate node based on its residual energy and association to the neighborhood. Moreover, Welcome enables nodes to auto-organize where eligible nodes content to become delegate in the absence of an existing delegate node.

Previously mentioned context-aware Publish/Subscribe systems focus mainly on location and time-stamp-based filtering, while context can be a broader notion than simple location and time-stamp information. In this direction, filtering-out data close to data sources based on “Context at the edge” could ensure more accurate data collection and at the same time

avoid useless information transmissions. Our last work is a Context-aware Publish/Subscribe that filter-out data near to data sources based on complex context events. Hence, we filter more data and avoid much more unnecessary data transmissions which could save more energy reduce overhead and increase the throughput. IoT devices in the same location provide information about different aspects of the environment, thus, we can leverage their coexistence to infer their respective contexts. Precisely, we propose to measure the relation of a device with the nearby devices with the goal to capture the amount of information they can provide regarding each other environments. Afterwards, any IoT device can take the decision to sign a “contract” with some of its neighbors based a dynamic coalitional game [48] for energy-constrained devices. Objects can merge into or split from spatio-temporal coalitions based on the payoff each coalition provides, in order to facilitate an energy efficient and high quality data collection.

2.7 Conclusion

Thanks to the great amount of connected objects increasing every day, it is now becoming possible to perceive events and changes within the ambient space surrounding users and adapt application behavior based on that. In this chapter, we reviewed ongoing research work tackling Context definition, neighbor discovery and context-aware Publish-Subscribe. However, existing context definitions considers only the context of a user either it is a person or a thing, to adapt the behavior of the application autonomously and enhance the user’s well-being. We highlighted the weaknesses of these definitions and suggest a new paradigm ”T2T context-awareness at the edge” that aims to expand the context-aware interactions from the application front-end to the application back-end. We aims to enable IoT devices/services to sense, analyze and understand their own as well as mutual situational context and adapt their behavior based on, without any human intervention. This is to be achieved in order to best preserve IoT devices resources while enhancing the quality of service. Accordingly, we propose three innovative IoT middleware components to give IoT devices this layer of intelligence: a context reasoner, a low latency and energy efficient neighbor discovery scheme and an energy-efficient and context-aware Publish-subscribe.

Chapter 3

Smart things: T2T

Context-awareness at the edge

Reasoner

Contents

3.1 Introduction	24
3.2 Background: Conditional Random Field	26
3.3 CRF for Context Awareness at the Edge	28
3.3.1 System Model	28
3.3.2 Find Co-existing Objects Influence Relationship	29
3.4 Performance Evaluation	36
3.4.1 Simulation over a synthetic dataset	37
3.4.2 Simulation over a real dataset	43
3.5 Conclusions	44

3.1 Introduction

Internet of Things (IoT) today comprises sensors, actuators and other connected devices enabled to provide different services in order to facilitate our lifestyle. It is now possible for an IoT application to find different IoT candidates deployed to collect data in an urban environment. However, the existence of hundreds of interconnected devices make it difficult

to find the appropriate sensor matching a user request [53] [54] [27]. At the same time, the abundance of data collectors allows the emergence of a new era of context aware applications, enabling such applications to adapt their behavior to the environmental conditions in which they are operating. Several definitions of “context” have been proposed and the most widely used is the one given by Dey et.al [14].

The IoT applications rely on sensors designed to detect and report different physical events which are typically installed in uncontrolled environments. For instance, a temperature sensor is designed to sense the temperature, though, its context (whether it is installed indoor/outdoors, in ambient/in shade, on a human body) can influence the service it provides. Temperature readings from two sensors deployed in the same location, one in ambient sunlight and the other under shade will result in different readings influenced by the physical conditions of their surroundings, i.e. temperature feelings under shade is different than the one felt in ambient sunlight.

In this chapter, we address the need for smart things context and we propose “*Context Awareness at the Edge*” reasoner as a dynamic annotation tool. Henceforward, IoT things may become conscious of their surroundings while gathering data/acting in the environment. The main purpose is to allow a device to adapt its behavior given its context (precisely physical environment), while assisting an IoT application. In order to avoid using dedicated resources for this task we benefit from the collective intelligence between nearby sensors in the same area to serve this task. We believe multiple sensors in the same location but with different abilities can provide each other an amount of information about their respective contexts. As such a solution is based on an opportunistic network, where nodes are asynchronous and distant from each other, we need to further analyze these spatio-temporal disparities. Despite this fact, physical phenomena are continuous and spread over time and space, therefore they present a strong spatio-temporal dependency. Thus, there exists a strong correlation between observations from neighboring sensors and the physical state of an IoT device. In turn, it is important to consider both, spatial and temporal relationships between IoT devices, in order to accurately estimate their contexts.

In this work, we use Conditional Random Field [52] to find the accurate spatio-temporal relations between different IoT devices. CRF is a probabilistic model for structured prediction widely used in natural language processing, computer vision, and bioinformatics. We adapt it here for an IoT device to model its context by learning the spatio-temporal relations with nearby sensors which are inversely proportional to the time difference as well as the distance between them. Similarly, it captures the relation between the device's current and previous states namely interactions. The existence of such spatio-temporal relations is validated by implementing the proposed CRF-based model using synthetic trace based simulations as well as real-testbed. Results show that it estimates the context of IoT devices with a high accuracy (up to 98.5%). The results also demonstrate the effect of spatio-temporal difference between readings from different nearby sensors on the estimation context accuracy. The contributions of this work can thus be summarized as follows:

- A novel concept of **“T2T context-awareness at the Edge”** is proposed giving sensors a new dimension of intelligence by allowing them to be aware of their own context.
- A Conditional Random Field based solution to accurately estimate, for an IoT device, the sequence of contexts it passes through during its "lifetime".
- A simulation study, comprising study both on synthetic and real-world data-sets, this simulation study allows to assess our CRF-based solution for context-awareness at the Edge.

The rest of the chapter is organized as follows. The next [Section 3.2](#) discusses the related work and background of Conditional Random Field. The [Section 3.3](#) explains our proposed T2T context awareness at the Edge model based on CRF, followed by performance evaluation and results discussion for both synthetic and real-dataset in [Section 3.4.1](#) and [Section 3.4.2](#), respectively. The [Section 3.5](#) summarizes our proposed model weakness and efficiency and present some insights into future directions.

3.2 Background: Conditional Random Field

Sequence labeling is a common issue in several domains, where data can exhibit sequential, temporal, spatial or spatio-temporal dependencies. Conditional Random Field [55] is a

probabilistic graphical model for sequence labeling, that takes advantage from unstructured learning approaches by considering correlations between labels. CRF estimates globally conditioned on the whole observation sequence X , the most likely label sequence Y (a label is class to predict, i.e. an instance of context to estimate). Let us consider $G = (V, E)$ a graph where each label y_i is encoded by a node $v_i \in V$ and the conditional (in)dependence between any two labels is represented by a vertex $e \in E$ linking their respective nodes. The joint distribution (Y, X) is a CRF if each $y_i \in Y$ obeys the Markov property such that the distribution $P(y_i|X, Y_{\forall j \neq i})$ depends only on its neighbors $P(y_i|X, Y_{\forall j \in N_i})$ according to the graph G . Sequence labeling allows us to introduce prior knowledge regarding the data dependencies based on potential functions where the potential functions can be of two types: (i) $\psi_i(y_i, X)$ is called association potential function to capture the degree to which a label y_i of the sequence can be assigned to the set of observations X . Similarly, (ii) $\psi_{ij}(y_i, y_j, X)$ is called interaction potential function, designed to encode our prior belief about relationships between neighbor's labels and the whole observation sequence X . Each potential function is a weighted-sum of feature functions, and we can use as many functions as needed to capture observations-labels relations. Thus, the joint distribution over the label sequence Y given X has the form:

$$p(Y|X) = \frac{\prod_{y_i \in Y} \exp(\psi_i(y_i, X) + \psi_{ij}(y_i, y_j, X))}{\sum_Y \prod_{y_i \in Y} \exp(\psi_i(y_i, X) + \psi_{ij}(y_i, y_j, X))}$$

Spatio-temporal extensions of the basic CRF model has been proposed. The main proposed extensions are made by Hoberg et al. [56] and Kenduiywoa et.al [57] for remote sensing data where both works introduced spatial and temporal interaction terms. In sensor networks, observations of the environment (e.g. temperature, and luminosity) exhibit a spatio-temporal correlation as such physical conditions rarely occur independently, thereby spatio-temporal CRF provide a promising solution. However, prior works only consider spatial where the associations predict labels only based on observations at the same timeslot. We propose to extend the model by spatio-temporal association terms to take into account spatio-temporal observations-label dependencies. Thus, in order to accurately estimate the context, we suggest to consider the observations from co-located sensors. Moreover, in sensor networks, due to time synchronization problems or network failures, observations belong rarely to the same timeslot and due to the mobility of nodes there is a dynamic spatial relation between an IoT device and the sensors reporting its environment. We model

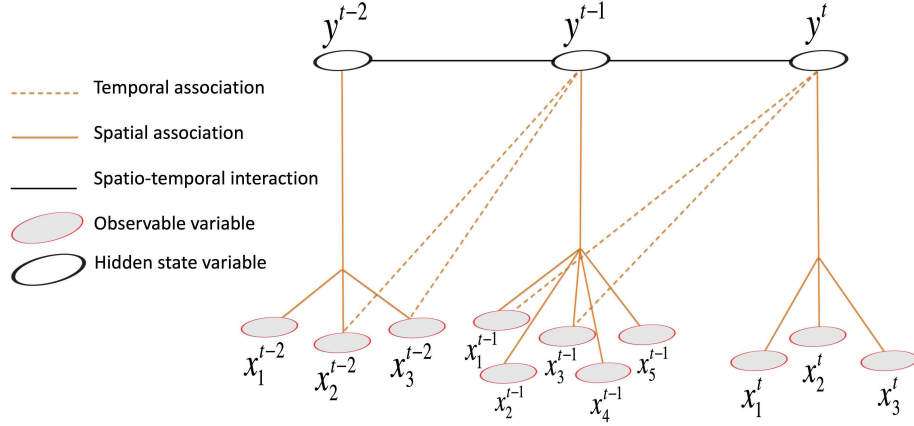


Figure 3.1: IoT spatio-temporal chain CRF Model vs spatio-temporal chain CRF model

the spatial and temporal associations separately as the lagtime and the distance can have different impacts on context estimation depending on the application domain. Hence, spatial and temporal associations can be derived based on observations as well as the spatial and the temporal distance vectors. For instance, the [Figure 3.1](#) models a spatio-temporal chain CRF with temporal and spatial associations terms modeled separately with orange continuous and dashed lines, respectively. In the following section we describe how we use CRF along with spatio-temporal relationship between the IoT devices and the environment observations to derive the context sequences they passed through.

3.3 CRF for Context Awareness at the Edge

In this section we formally describe the CRF based model for context awareness at the Edge where sensors are made aware of their surroundings. We define below the network model to consider through the remainder of the work.

3.3.1 System Model

3.3.1.1 Network spatio-temporal parameters

We consider a time varying network modeled as an undirected graph $G(V(t), E^v(t))$, where $V(t) = \{v\}$ is a set of vertices v , each representing a device/node at time t . $E^v(t) = \{e_{ab}(t) \mid v_a, v_b \in V, a \neq b\}$ is the set of edges, $e_{ab}(t)$ modeling the existence of a direct communication link between nodes v_a and v_b at time t . The number of edges for a node

are the number of co-located nodes or neighbors connected to it at time t (temporal) as well as geographically situated within its communication range (spatial). Furthermore, the distance between two nodes v_a and v_b is represented by the Euclidean distance $d(v_a, v_b)$ between their respective GPS coordinates. To consider a temporal network characteristics, the time T is divided into uniform time-slots $T = (t_1, t_2 \dots)$ where an IoT device $v \in V$ periodically reports its sensory data readings after a time-slot t and different sensors can report readings at different time-slots.

3.3.1.2 CRF key parameters

We define the set $X = \{X^t\}, \forall t \in [1, |T|]$ to represent the whole observations or readings captured by the sensors (IoT devices), available at a given neighborhood, for different phenomena such as temperature, luminosity etc, at a time period T . Where $X^t = \{x_s^t\}, \forall s \in [1, |V(t)|]$, is the set of observations provided by the co-existing sensors at timeslot t . We consider $X^{t-1} = \{x_r^{t-1}\}, \forall r \in [1, |V(t-1)|]$ as the set of observations provided by the co-existing sensors at timeslot $t-1$. Similarly, $Y = \{Y_s^t\}, \forall t \in [1, |T|]$ is the set of states (context) each object s passed through during the period of Time T . Where the state can be the representation of its physical condition or environment. For example, based on an observation from a temperature sensor, we can infer its state as indoor/outdoor, under shade/sunlight, etc. Thus, hereinafter, we consider a spatio-temporal network model where we analyze the relation between different states Y inferred from the observations X for a set of devices V . It is to note that we will refer to object, sensor, thing, IoT device or node interchangeably in the remaining of the article, thus representing the same entity.

3.3.2 Find Co-existing Objects Influence Relationship

The dynamic environment particularly the mobility of nodes makes it difficult to identify the impact of the environment on a device. We assume that the environmental reports from nearby sensors can be altered by their freshness and spatial distance therefore, we capture indirect influences between co-existing sensors in two ways: (i) *Temporal*: where multiple devices reporting their observations at different time differences the more recent ones will have greater influence on the context estimation, and (ii) *Spatial*: in order to estimate the context, the neighboring devices closer with respect to their distances to the object under study can have relatively more influence compared to the ones farther away. Similarly, we

Table 3.1: List of Notations

Notation	Description
$V(t) = \{v\}$	Set of objects/nodes at timeslot t
$E^v(t)$	Set of Edges between objects at timeslot t
$X = \{x\}$	Set of observations for a set of objects in a vicinity
$Y = \{y\}$	Set of states/labels/contexts
T/t	Total time/time-slot
V_i	set of neighbors of object i
$V_{i,T}$	Set of temporal neighborhood of object i
$V_{i,S}$	Set of spatial neighborhood of object i
$X_v^t = \{x_{q,v}^t \mid q \in 1..l\}$	Set of observations of neighbor v at timeslot t
$X_t = \{X_v^t \mid v \in V_{i,S}\}$	Set of neighborhood's observations of object i at timeslot t
$X_{t-1} = \{X_v^{t-1} \mid v \in V_{i,T}\}$	Set of neighborhood's observations of object i at previous timeslot $t - 1$
x_i^t	Observation of object i at timeslot t
x_i^{t-1}	Observation of object i at timeslot $t - 1$
Y_t^i	Set of contexts, the object i passed through during timeslot t
y_i^t	Context label at time t
y_i^{t-1}	Context label at time $t - 1$
K	Set of spatial neighbors feature functions
N	Set of temporal neighbors feature functions
L	Set of interaction feature functions
ψ_t/ψ_s	Temporal/Spatial Association Potential
f_j/f_k	Temporal/Spatial feature function
ψ_{ij}	Spatio-temporal interaction potential
g_t	Spatio-temporal feature function
$d_t(i, j)$	Distance between two nodes s, r
λ_j	Temporal association weighting parameter
λ_k	Spatial association weighting parameter
β_t	Spatio-temporal interaction potential parameter
Δ_T	Time difference for neighbor observations
Δ_S	Location difference for neighbor observations
d/t	Mean Euclidean Distance to the neighbor's observations
t	Mean Time decay to the neighbor's observations
d_{th}/t_{th}	Threshold distance/time
δ_d/δ_t	Tuning parameter spatial/temporal
$Z(x)$	Normalizing constant/partition function

can predict the physical state of an object from its previous state. A temperature sensor physically located in an ambient environment is more likely to provide similar readings between two close time instants, if the observations of its environment remain the same. Therefore, we use CRF to model the inter-object relation for the context awareness.

3.3.2.1 Temporal Association Potential

A temporal association potential ψ_t is dedicated to capture how likely a sensor v_i takes context label $y_{i,t}$ at the timeslot t , given its temporal neighbors observations $X_{t-1} = \{X_v^{t-1} \mid v \in V_{i,t}\}$. The Figure 3.2 presents the Temporal association relations (dashed orange lines) between a hidden state variable (context) to estimate and the associated observations from nearby sensors at previous timeslots.

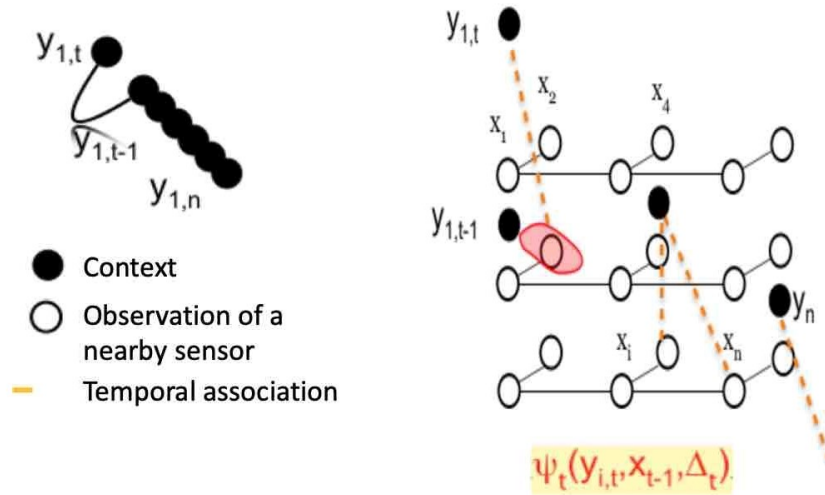


Figure 3.2: Temporal associations

It is possible that a clock drift between different deployed sensors in a neighborhood results in asynchronous observations by each device. However, in a dynamic environment, relatively large time differences between sensors can impact their respective context estimation. To cater such a lag between a sensor and its nearby sensors readings, the temporal association potential ψ_t is derived using a time decay function given as:

$$\psi_t(y_i^t, X_{t-1}, \Delta_T) = \sum_{j \in N} \lambda_j f_j(y_i^t, X_{t-1}, \Delta_T), \quad (3.1)$$

where the impact of the temporal neighbors $V_{i,T} \subset V$ on the context estimation y_i of node v_i , at current time-slot t is characterized by a set of feature functions f_j based on the neighborhood observations X_{t-1} from previous to current time-slot and their temporal distance Δ_T . In this context f_j can be based on any probabilistic classifier (e.g. random forest) [55]. The vector $\Delta_T = [t_{i,1}, t_{i,2}, \dots, t_{i,nt}]$ represents the time differences between the observation of v_i and each of its temporal neighbors $v \in V_{i,T}$. The parameter λ_j is a weighting parameter for the corresponding feature function f_j , to be learned by the model based on the time sensitivity requirements of an object's neighborhood. One simple feature function for the temporal association potential can be defined as an exponential decay function where the impact of a neighbor decreases with time:

$$f_j(y_i^t, X_{t-1}, \Delta_T) = P_j(y_i^t | X_{t-1}) e^{-\delta_t t_{V_{i,t}}}, t_{V_{i,t}} \leq t_{th} \quad (3.2)$$

where, $P_j(y_i^t | X_{t-1})$ is a binary real-valued function to compute the potential sensor's context taking a value y_i^t . Where X_{t-1} can refer to readings from one to many neighbors. $t_{V_{i,t}}$ is the lag time (mean lag times) between the sensor under study v_i , and the neighborhood observation(s) X^{t-1} considered to this task. Only observations in the earlier times (\leq time threshold t_{th}) are selected. $0 \leq \delta_t$ is the tuning parameter to adjust the effect of temporal distances between observations. Thus, if this time is earlier than a certain threshold time specified t_{th} , the feature function follows an exponential decay.

3.3.2.2 Spatial Association Potential

A Spatial association potential ψ_s is dedicated to capture how likely a sensor v_i takes context label $y_{i,t}$ at the timeslot t , given its spatial neighbors observations $X_t = x_{i,t}, x_{n,t}$. The [Figure 3.3](#) presents the Spatial association relations (Continuous orange lines) between a hidden state variable (context) to estimate and the associated observations from nearby sensors at the current timeslot.

Besides temporal association, it is natural to consider the impact of the sensors/neighbors with respect to their distance. The idea is to penalize further neighbors and their

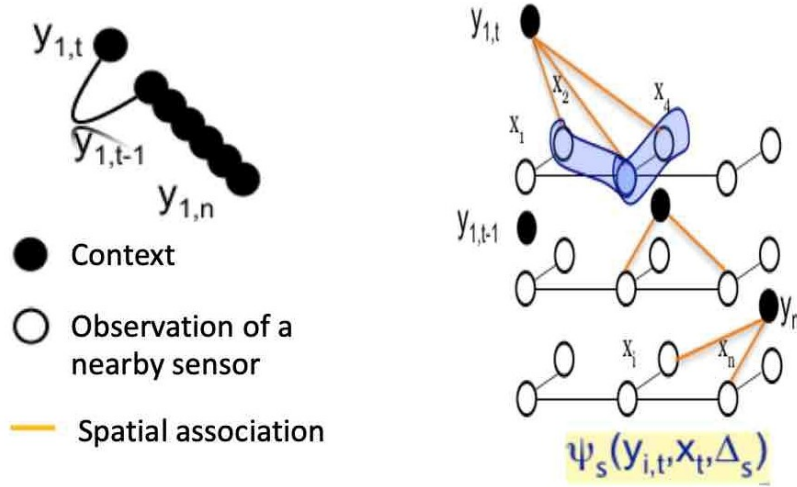


Figure 3.3: Spatial association

respective observations. Therefore, we define the Spatial Association Potential ψ_s to capture, for a node v_i , the impact of its spatial neighbors based on their distance as:

$$\psi_s(y_i^t, X_t, \Delta_S) = \sum_{k \in K} \lambda_k f_k(y_i^t, X_t, \Delta_S), \quad (3.3)$$

Where, f_k is the feature function representing the spatial correlation between the observations from neighboring sensors on the state y_i of the node v_i . For a set of spatial neighbors $V_{i,S} \subset V$, $X_t = \{X_v^t \mid v \in V_{i,S}\}$ is the set of spatial neighbors observations at time-slot t . The term $\Delta_S = [d_{i,1}, d_{i,2}, \dots, d_{i,n_s}]$ indicates the distances between the sensor v_i and each of its spatial neighbors. The term λ_k is a spatial parameter to be learned by the model, it measures the impact of neighbor's relative distance which can be application domain dependent. There exist different possibilities for such a feature function to penalize large distances between object. We use the principle of inverse distance relation represented as:

$$f_k(y_i^t, X_t, \Delta_S) = P_k(y_i^t | X_t) (d_{V_{i,s}} + 1)^{-\delta_d}, d_{V_{i,s}} \leq d_{th} \quad (3.4)$$

The Spatial feature function in Equation (3.4) is a function of the inverse of the euclidean distance between the node v_i and where the location(s), the observation(s) X_t , are taken by the node(s) in the vicinity $V_{i,S} \in V$. However, a threshold distance $0 \leq d_{th}$ should be respected. For instance one should imagine a smart home where only readings from sensors located within the house boundaries are considered. In case we are interested in association

from sensors which are located indoors, the feature function can define d_{th} within a range of positions corresponding to indoor locations only. The tuning parameter $1 \leq \delta_d$ controls the extent of the spatial distance impact which can be defined according to the applications requirement.

3.3.2.3 Spatio-temporal Interaction Potential

So far we discussed temporal and spatial relations to the observations of other nearby sensors. An object can also be influenced by its own spatio-temporal physical state. Precisely, its physical conditions are likely to stay the same if it remains at the same previous location, since the physical conditions evolve gradually. To model such a joint impact, we define the term ψ_{ij} as the Spatio-temporal interaction potential, modeled by continuous black lines in Figure 3.4, to capture the degree to which y_i^t , the state of the node v_i at time t , is correlated to the previous state y_i^{t-1}

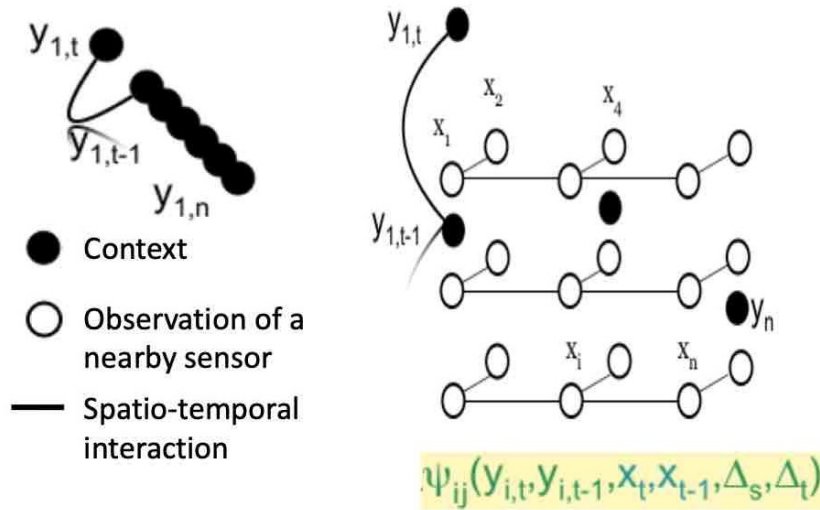


Figure 3.4: Spatio-temporal interaction

We define the Spatio-temporal interaction term ψ_s to capture, for a node v_i , the impact of its spatial neighbors based on their distance as: as follows:

$$\psi_{ij}(y_i^t, y_i^{t-1}, X_t, X_{t-1}) = \sum_{l \in L} \beta_l g_l(y_i^t, y_i^{t-1}, X_t, X_{t-1}) \quad (3.5)$$

where, L is the set of spatio-temporal interactions between its own physical conditions. For a sensor v_i , Equation (3.5) considers the spatio-temporal relations between two consecutive sensor's states y_i^t, y_i^{t-1} based on possible spatio-temporal correlations among their respective nearby sensor reading X_{t-1}, X_t . Feature function g_l is based on similarity or dissimilarity measures to estimate the node's own state at the current and previous time, thus, reflecting a temporal relation with its own physical conditions. The term β_l is an interaction parameter to adjust the spatio-temporal impact of the node transition, due to its mobility or changes in the environment. Here g_l is the effect of the spatio-temporal influence from both the sensors own previous states and the observations from nearby sensors:

$$g_l(y_i^t, y_i^{t-1}, X_t, X_{t-1}) = P_l(y_i^t | y_i^{t-1}, X_t, X_{t-1}) e^{(\alpha_l t_i + \Gamma_l d_i)}, \quad d_i \leq d_{th} \ \&\& \ \delta_t \leq t_{th} \quad (3.6)$$

where, d_i is the distance of the node v_i to its previous spatial neighbors reading, $0 \leq d_{th}$ is the threshold distance specified, as the previous neighbors can no longer be considered due to the high mobility. Similarly, t_i is the time difference between the set of observations X_{t-1} and the current time slot, where $0 \leq t_{th}$ is the threshold time specified for the temporal association function to meet a certain accuracy requirement. $0 \leq \alpha_l$ and $0 \leq \Gamma_l$ are the spatial and temporal tuning parameters to adjust the spatial and temporal distances effects. The spatio-temporal feature function for the interaction potential considers both, the spatial and temporal observations near a sensor as well as the sensor's previous state.

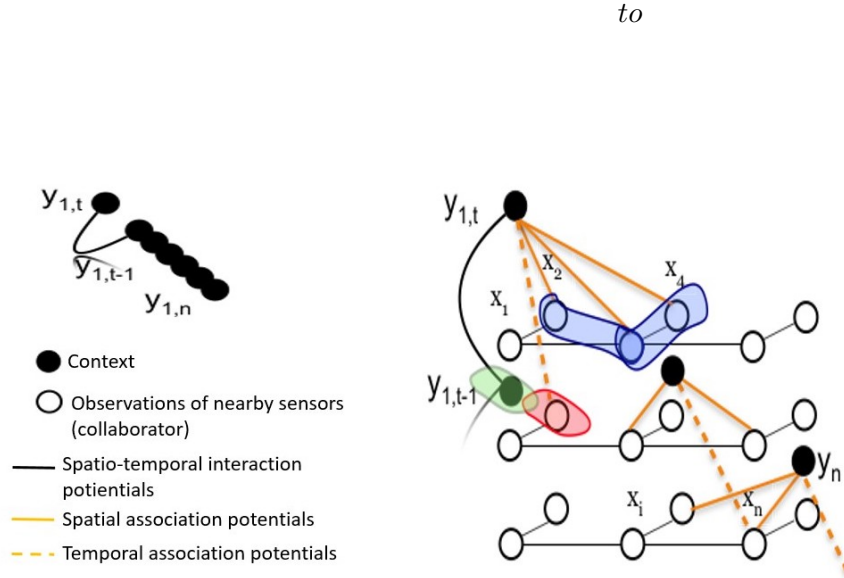
3.3.2.4 Spatio-temporal CRF

The conditional probability between the spatial and temporal associations based on the observations, and the spatio-temporal interactions is given as:

$$p(Y | X) = \frac{1}{Z(Y)} \prod_{y \in Y} \prod_{t \in T} \exp(\psi_s(y_i^t, X_t, \Delta_S) + \psi_t(y_i^t, X_t^{t-1}, \Delta_T) + \psi_{ij}(y_i^t, y_i^{t-1}, X^t, X^{t-1}, \Delta_S, \Delta_T)), \quad (3.7)$$

Where,

$$Z(Y) = \sum_Y \prod_{y \in Y} \prod_{t \in T} \exp(\psi_s + \psi_t + \psi_{ij}), to \quad (3.8)$$



$$\mathbf{P}(\mathbf{Y}|\mathbf{X}) = \frac{\prod_{y_i} \prod_t \exp\{\psi_s + \psi_t + \psi_{st}\}}{\sum_Y \prod_{y_i} \prod_t \exp\{\psi_s + \psi_t + \psi_{st}\}}$$

Figure 3.5: Spatio-temporal IoT CRF model

ψ_t and ψ_s are the spatial and temporal association potentials between the node and the observations X_{t-1} and X_t of a set of temporal neighbors $V_{i,T}$ and a set of spatial neighbor $V_{i,S}$ respectively. Similarly, ψ_{ij} is the spatio-temporal interaction potential for a node v_i between its previous state y_i^{t-1} and the current state y_i^t . The term $Z(Y)$ is a normalizing constant called partition function. Thus, Figure 3.5 allows to model how a sequence of context can be estimated based on the node's context taking into account changes in the environment. In particular, the time lag between sensor readings and changes in locations. The proposed model penalizes large difference in time and distance between co-existing sensors as well as time or distance difference between in a sensor own physical conditions.

3.4 Performance Evaluation

We evaluate the performance of the proposed CRF based model on synthetic data as well as experiments on real-world dataset. The objective is to find out how much accuracy is reached by our model to estimate the context of an object. We show that it is feasible to learn an object's context only by using the data collected from its nearby sensors. We discussed below our simulation study followed by experimentation evaluation.

3.4.1 Simulation over a synthetic dataset

We consider a temperature sensor as the object v_i under study to infer its context sequences $\Theta = \{Y_1, Y_2, \dots, Y_M\}, Y_k = \{y_i\}, \forall i \in [1, N]$. The estimation of each label sequence is based on the observations of its vicinity provided by the nearby sensors under its communication range (100 meters). The output sequence is defined for N consecutive readings of the sensor v_i , over the limited set of possible contexts $y_i \in \textit{indoor}, \textit{outdoor ambient}, \textit{outdoor under shade}$. We limit the types of nearby object to temperature and luminosity sensors as both provide sufficient information about object's context. Each sensor's reading is annotated by GPS and timestamp. In order to simulate a real scenario, we consider asynchronous sensors which results in lagtimes between different neighbors readings. The spatial distance between nodes is based on Euclidian distance between their GPS coordinates. We compute the time difference in seconds for each pair of sensor readings. It is to note that in the case of several readings of a neighbor belong to the same timeslot we just consider the most recent reading.

3.4.1.1 Potential Functions

The association functions allow to introduce prior knowledge on the output/input dependence. The input could be a set of one to many neighbors observations. In our experimentation we consider nine associations feature functions listed in the [Table 3.2](#). For instance, on a sunny day the luminosity under shade is about 10 times intenser than an overcast day, thus the same period of day (day time) and the same context (under shade) can correspond to different luminosity ranges depending on the weather conditions, i.e. $Undershade = \{Sunny = [1000, 2000]lux, Overcast = [100, 200]lux\}$.

For instance, let's consider two association feature functions, the first one associates the ambient context for a sunny day at La Rochelle in a day period to an interval of luminosity $[1000, 2000]$ lux, [Equation \(3.12\)](#). The second feature function [Equation \(3.13\)](#) associates ambient context for a overcast day at La Rochelle in a day period to an interval of luminosity $[100, 200]$ lux. Functions f1 and f2 can be described as follow:

$$f_1(\text{ambient}, L^t, \Delta_S) = \begin{cases} (d_{V_{i,s}} + 1)^{-\delta_d} & L^t \in [1000, 2000]lux \ \&\& \ d_{V_{i,s}} \leq d_{th} \\ 0 & \text{otherwise} \end{cases} \quad (3.12)$$

$$f_2(\text{ambient}, L^t, \Delta_S) = \begin{cases} (d_{V_{i,s}} + 1)^{-\delta_d} & L^t \in [100, 200]lux \ \&\& \ d_{V_{i,s}} \leq d_{th} \\ 0 & \text{otherwise} \end{cases} \quad (3.13)$$

The interaction feature functions can be described as a combination based on dissimilarities and similarities between sensor previous and current states. The dissimilarity based interaction feature function can be considered for the case when the previous luminosity at $t - 1$ as $\{[10000, 20000], [100, 200]\}$ is increased at t to $\{[70000, 110000], [10000, 20000]\}$. We can infer that the sensor state is changed from $y_{i-1} = \text{under shade}$ to $y_i = \text{outdoor ambient}$. Similarly another dissimilarity based interaction feature function is when the temperature difference between t and $t - 1$ is about -5 up to -10 degrees and the luminosity at $t - 1$, $\{[70000, 110000], [10000, 20000]\}$ is decreased at t to $\{[10000, 20000], [100, 200]\}$, thus, we can estimate that the sensor state $y_{i-1} = \text{ambient}$ is changed to $y_i = \text{under shade}$, as described in interaction function [Equation \(3.14\)](#). For the similarity based interaction feature function, if both x_{t-1} and x_t are within the same range, we can infer that $y_i = y_{i-1}$, the sensor context remains the same.

$$g_1(\text{ambient}, \text{under - shade}, T^t, T^{t-1}, \Delta_S T) = \begin{cases} (e^{(\alpha_t t_i + \Gamma_1 d_i)}) & T^t - T^{t-1} = 5 \\ \& \ d_i \leq d_{th}, \delta_t \leq t_{th} \\ 0 & \text{otherwise} \end{cases} \quad (3.14)$$

3.4.1.2 Synthetic data set

As there is no simulator of data able to annotate the generated data with its physical context, we build our own simulator. For this task, we use five SensorTag IoT kits (CC2650STK

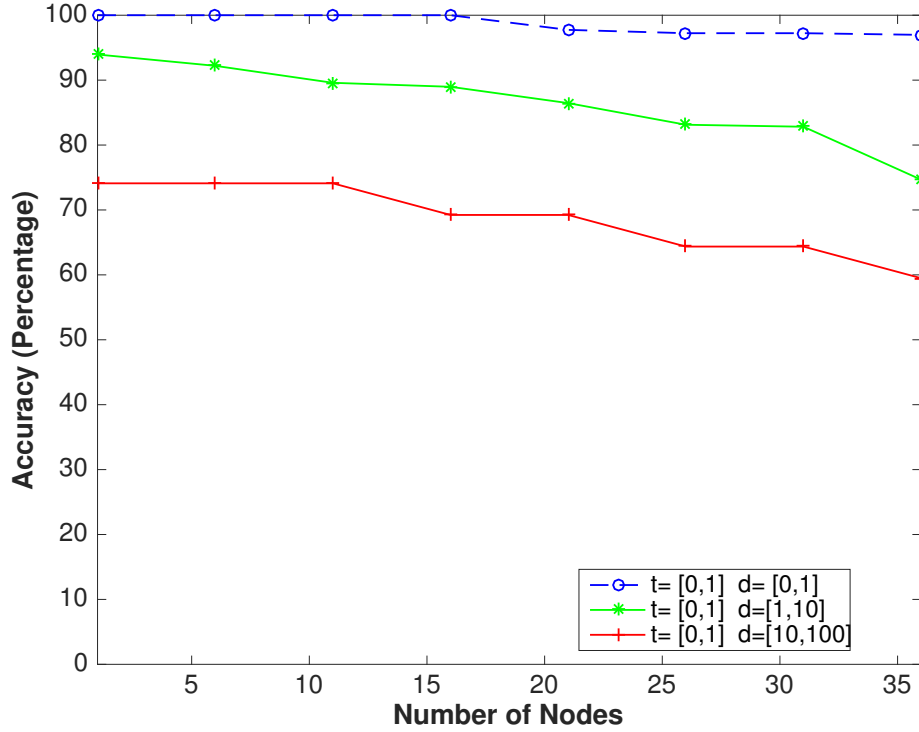
Table 3.2: associations feature functions

Feature functions	Period of Day	Temperature	Location	Luminosity range (lux)	Context
Sunny day outdoor noon	Day	18 < Hot < 40	outdoor	70000-110 000	Outdoor ambient
Overcast day outdoor noon	Day	-40 < Cold < 18	outdoor	1000-2000	Outdoor ambient
Sunny day shadow noon	Day	18 < Hot < 40	outdoor	10000-20000	Under shade
Overcast day shadow noon	Day	-40 < Cold < 18	outdoor	100-200	Under shade
Sunny day dusk	Dusk	18 < Hot < 40	outdoor	200-400	Outdoor ambient
Overcast day dusk	Dusk	-40 < Cold < 18	outdoor	20-40	Outdoor ambient
Sunny day night	Night	-	outdoor	0.0001-0.00015	Outdoor ambient
Overcast day night	Night	-	outdoor	0.0002-1	Outdoor Ambient
Indoor	-	-	indoor	100-500	Indoor

SimpleLink Bluetooth low energy/Multi-standard SensorTags) each equipped with 10 sensors including support for light, digital microphone, magnetic sensor, humidity, pressure, accelerometer, gyroscope, magnetometer, object temperature, and ambient temperature. We connect each of the deployed SensorTag to a GPS sensor and a raspberry PI. We deploy IoT kits at the campus of University and at a home in La Rochelle, France, with different periods of the day(day, dusk, night), different weather conditions(sunny,cloudy) and different context conditions (under shade/ ambient, indoor/outdoor). The data collected for hours results in a reference real world data set, that we used to extract some prior knowledge about the correlation of some physical phenomenas(temperature, luminosity) and the desired contexts (indoor, ambient, under shade). For instance we note that luminosity around an IoT object exposed to ambient sunlight lies between 70000 and 110000 lux for a sunny day in day period. Based on the extracted correlations we implemented a simulator able to generate a set of data for a/some sensors, given some input features like the physical context, the weather condition, the period of day, and the frequency of collect.

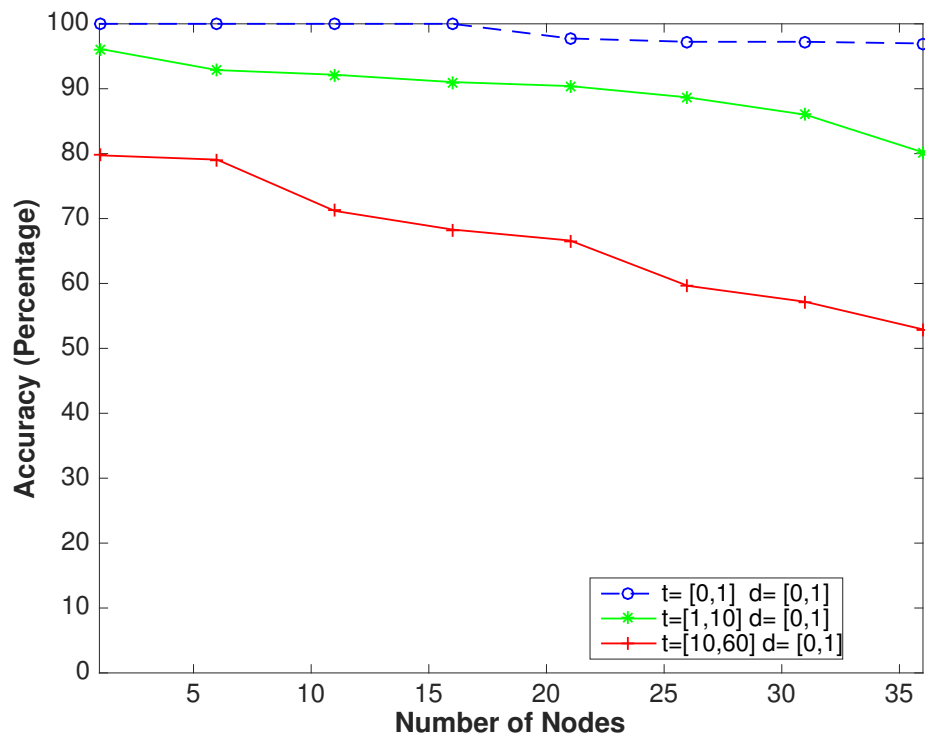
3.4.1.3 Simulation scenario based a synthetic dataset

We simulate a network with 36 sensortags each one composed by a luminosity, temperature and GPS sensors. The sensortags are mobile in an area of $100 m^2$. We Collect data with different frequencies during 30 hours. We consider different (sunny and overcast) weather conditions and periods of the day (day, dusk, and night). The distances between sensors denoted by $d_{i,k}$ can vary randomly within the following (spatial) ranges $[0, 1]$, $[1, 10]$ and $[10, 100]$ and measured in meters. We select $M = 150$ label sequences of size $N = 10$ corresponding to 10 consecutive 5-second time-slots of a given sensortag, to annotate with

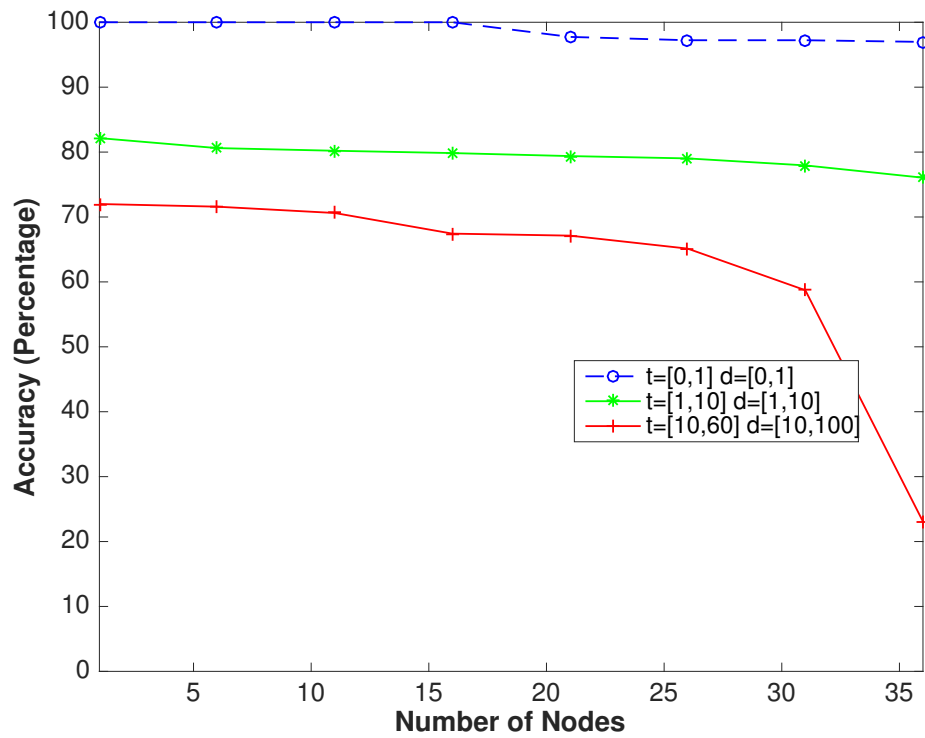


(a) Varying nodes and distance (t=0,1)

context. We train the model on $M = 90$ label sequences. For each sequence we select as input $10 * 10 = 100$ nearby sensor readings with different lagtimes and different distances from the object under study. The model is trained on the above described training set, using limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), an optimization algorithm in the family of quasi-Newton methods [58] to find the set of weighting parameters which maximizes the likelihood. The proposed CRF-based model is evaluated using accuracy as the performance metric. The accuracy of the model is obtained by summing the number of true positive and true negative label occurrences divided by the whole output set ($M * N$ labels). We test the model on $M = 60$ label sequences, with different input data sets varying from 100 up to 3600 input readings, corresponding to different neighborhood size from 5 to 36 sensortags. We vary input data properties in terms of source distance and up-to-dateness to study the marginal and joint effect of each criteria on the accuracy of our model. Each testing process is repeated 10 times such that model is tested with 10 folds. Finally average testing overall accuracy is calculated. The tuning parameters are application dependent, some tests can help to choose the best parameters. We choose, in the following, to set δ_t and δ_d to 1 and each of α_l and Γ_l to 0,5.



(b) Varying nodes and time (d=0,1)



(c) Varying nodes, distance and time

Figure 3.6: Varying nodes vs distance and time

3.4.1.4 Testing: Impact of increasing sensors (Simulations)

Section 3.4.1.3 shows the results of the effect on the accuracy by increasing the number of neighboring nodes for different input data properties. In Figure 3.6(a), the time between readings for the sensors is kept in the range of $[0, 1]$ minutes while the distances between them are varied in three ranges $[0, 1]$, $[1, 10]$ and $[10, 100]$ meters. The best accuracy is observed when the sensors are spatio-temporally close (time in the range $[0, 1]$ seconds and distance in the range $[0, 1]$ meters). In this case, our model predicts the sensor states with more than 95% accuracy. We observe that the accuracy decreases with the increase of the inter-object distance in the range $[1, 10]$ and $[10, 100]$ meters, up to 75% and 60% accuracy can be achieved respectively, while the lagtime range remains the same. The further the sensors are from each other the less accurate the context of a sensor can be estimated. We can therefore, infer that the proposed CRF-based model yields a high accuracy in case there exists enough close neighbors as they can better observe the reference device's environment.

Similarly, Figure 3.6(b) shows the impact of the increasing age of observations along with their number on the model accuracy for sensors placed at similar distances from each other. We observe that for higher time difference between sensor readings, there is a substantial difference in the accuracy of the estimated context. Again, we observe that the accuracy decreases slightly with the increase in the number of co-located nodes with at least 96% for 36 nodes. An accuracy of 53% is observed for the case we deploy 36 nodes and the time differences of sensor readings are around $[10, 60]$ seconds, reflecting the worst case behavior. Compared to the variation of distance, previously discussed in Figure 3.6(a), a relatively high drop is noticed in the accuracy when the time is varied for sensor installed within $[0, 1]$ meter. Therefore, we can infer that increasing the lagtime of observations have more effect on the accuracy than increasing spatial distances, it can be due to the fact that some physical conditions (e.g. shade, sunlight) can cover a large area and consequently reported by distant neighbors. We also analyzed the impact of increase in the number of nodes while varying both, the time difference as well as the distances between nodes. We notice in Figure 3.6(c) that, in the worst case considered when both time differences and distances between sensors are increased (i.e. $t = [10, 60]$ and $d = [10, 100]$), the state of a sensor under study can be predicted with around 24% accuracy. Thus, from the overall analysis of the increase in number of sensors, the proposed model successfully predicted

the state of the sensor under study with high accuracy, even with the high spatio-temporal differences between their readings.

We can summarize that our CRF-model can perform well (up to 98.5% accuracy) in the existence of nearby neighbors reporting observation close to the sensor with respect to both distance as well as time. It is to note that lacking of any of the two conditions can affect the model accuracy particularly for devices operating in a rapidly changing environment.

3.4.2 Simulation over a real dataset

The dataset that we are using in our experiments consists of sensor readings recorded at the campus of University of La Rochelle, France. We deploy five SensorTag IoT kits (CC2650STK SimpleLink Bluetooth low energy/Multi-standard SensorTags) each equipped with a temperature and luminosity sensor. We collected data during 5 hours, alternating randomly for 10 time-slots, between three considered contexts. The deployed SensorTags collect data with different arbitrary lagtimes to the object under study [59s, 20s, 60s, 60s, 60s], which results in [1670, 4560, 1540, 1540, 1540] readings. We also vary GPS positions in the range of 100m from the device's under study, every thirty minutes. The sensors positions are randomly changed, each SensorTag is manually moved with a speed and direction chosen at random until we ensure a change in the environment conditions and/or vicinity. Consequently, the labels considered for this experiment are $Y = \{y_i\}, \forall y_i \in \{indoor, outdoor\ ambient, outdoor\ under\ shade\}$. We generate 200 sequences by randomly choosing for each sequence $N = 10$ consecutive time instants to estimate the context of a given object. For each selected time instant, we find the neighbors observations in the object's vicinity (i.e. within a threshold distance and time difference) as its spatial and temporal neighborhood.

We analyzed the accuracy of predicting the physical state of the sensor by finding the extent at which the age of context data should play a role. For all the experiments over real data sets, there are 2 sensors (temperature sensor, luminosity sensor) attached to the same IoT device we're interested in, that ensures for all context sequences to estimate there exist at least 2 partially accurate readings about the environment. We merge observations from these too close neighbors with readings from randomly positioned neighbors showed that varying the neighbor distances while there are 2 close neighbors results in very good

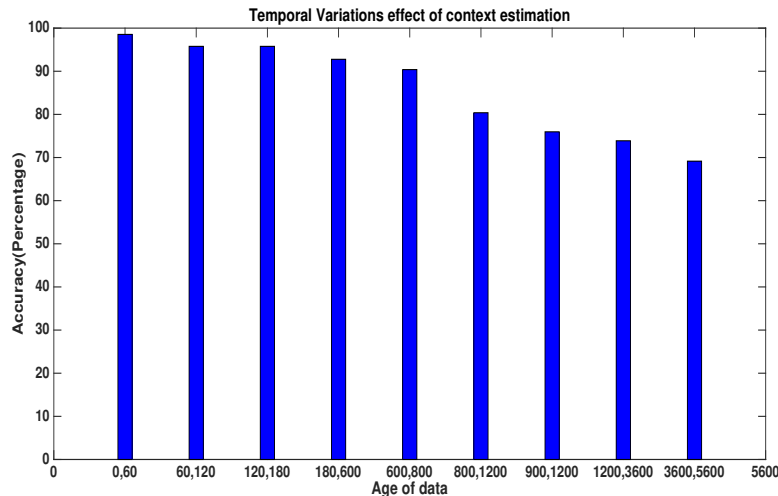


Figure 3.7: Effect of age of data variations on context estimation (Test-bed with 5 SensorTags)

results thanks to penalization functions that’s why we only focus on studying the impact of age of data variations. Figure 3.7 shows the impact on the accuracy of the increase in age of environment’s observations for a total of 10 sensor neighbors. We observe a decrease on the accuracy along with an increase of context data age. It is thus evident that the more the observations are farther in time, the less accurate they are. Still we observe around 98.5% accuracy for a lag time of 0 to 60 seconds, while we see around 69% accuracy for a lag time of $1h - 1h30m$. As already shown with experiments on synthetic data, real-world experiments also prove that our model is able to estimate an IoT device context deployed in an uncontrolled environment with high accuracy by just involving two types of information (temperature and luminosity) under certain spatial and temporal thresholds.

3.5 Conclusions

Devices providing services in an urban environment suffer from indifference regarding their context. For instance, such a lack of knowledge about the deployment context could result in the collection of inaccurate or sometimes useless and redundant data. Therefore, in this chapter we propose a context at the edge reasoner based a collaborative approach between devices in a neighborhood. Collective intelligence enables IoT devices to see, hear, and smell the physical world around without deploying any dedicated resources.

As physical phenomena are continuous, with rich spatio-temporal dependency, there exists a strong correlation between neighbor’s observations the physical state to estimate. In

order to capture such relations, we used Conditional Random Field to estimate the context of an Iot device based on the readings from its surroundings sensors and its own previous physical conditions. Thus, we adapt the spatio-temporal CRF existing models to consider separately the spatial and temporal potential functions, as they may have different impact in the context estimation. Experiments are performed on synthetic and real traces. Results showed that the proposed models successfully estimates the context of different sensors with high accuracy up to 98.5%. Obviously, the collaboration process requires nodes to detect neighbors within communication range and to connect with each other. IoT devices can easily move and change environment, they could enter and leave area/network at any time, therefore neighbor discovery is a continuous process. Thus, ensuring the accuracy and the continuity of context discovery service, comes down to ensuring for each node an updated network topology information. However, low-powered devices couldn't support energy hungry neighbor discovering to establish the topology information, therefore to efficiently connect and discover neighbor nodes for IoT devices is key prerequisite for any IoT architecture.

Chapter 4

WELCOME, Low Latency and Energy Efficient Neighbor Discovery for Mobile and IoT Devices

Contents

4.1	Introduction	49
4.2	Overview of Neighbor Discovery Schemes	50
4.3	WELCOME: Towards a Flock Discovery	53
4.3.1	System Model	53
4.3.2	Delegate Node Eligibility	54
4.3.3	Node activity schedule in WELCOME	55
4.3.4	Absence of WELCOME message - Self-organization	58
4.4	Analytical Evaluation	60
4.4.1	Evaluation Metrics	60
4.5	Theoretical Comparison	61
4.5.1	key parameter	61
4.5.2	Detailed theoretical analysis	62
4.6	Simulation-based Evaluation	66
4.6.1	Simulation Scenario	66

4.6.2 Simulation Results	68
4.7 Conclusions	73

4.1 Introduction

Energy efficient data collection requires low power sensors in a smart building/home to discover and communicate with nearby sensors and other wireless devices. Similarly, “T2T context-awareness at the Edge” could require wireless low-power devices in each others proximity to interact locally as an opportunistic network built upon sensors and IoT devices, such as tablets, smart phones and smart wearable. Due to the potential mobility of some IoT devices as well as other factors such as channel quality and devices capabilities, network nodes can leave and join network at any time thus opportunistic Neighbor discovery is a key concern for opportunistic and self-organized network. Neighbor discovery allows to dynamically trigger the interactions that might be needed, especially when building ”T2T context-awareness at the edge”.

Energy efficient neighbor discovery for multiple mobile devices in each other proximity is a challenge along duty cycling, where low power devices are inactive for a potentially large fraction of time. Existing schemes allow each device to employ a schedule to become active and send periodic messages or listen to neighboring devices to ensure a neighbor discovery in a bounded delay. However, collisions can occur due to simultaneous transmission of messages from multiple devices resulting in failure of neighbor discovery. So, our belief is that there is a need to reduce the number of message transmissions in a neighbor discovery process to avoid collisions and enhance the number of devices discovered.

To do so, in this chapter, we propose WELCOME, a low latency and energy efficient neighbor discovery scheme based on a single-delegate election algorithm. Instead of all nodes transmitting messages, only a single node at the time can become a delegate to discover the nodes in the vicinity and provide the neighborhood information to its neighbors. A node first finds its eligibility to become delegate based on its residual energy and its association to the neighborhood. It then declares itself a delegate and listens to messages from its neighbors. Finally, it broadcasts the information regarding its neighbors to the

devices in its communication range. Moreover, the delegate role can be rotated among neighbors where a node with high eligibility can contend to declare itself as the new delegate in case of absence of a delegate node. WELCOME is compared with seven existing neighbor discovery schemes and it successfully discovers 100% of neighbors with low energy consumption and low latency for a neighborhood size of up to 100 nodes.

The remainder of the chapter is organized such that the next [Section 4.2](#) presents an overview of neighbor discovery schemes. [Section 4.3](#) presents the design and description of our proposed WELCOME scheme. In [Section 4.4](#), we define the evaluation metrics along a theoretical comparison of WELCOME with the state of the art schemes. [Section 4.6](#) discusses the performance evaluation and results based on extensive simulations. Finally, [Section 4.7](#) concludes the chapter and provides some insights into future directions.

4.2 Overview of Neighbor Discovery Schemes

Stochastic-based schemes

Stochastic-based schemes allow nodes to transmit beacons, listen for beacons from other nodes or sleep in a slot based on a probability distribution.

Birthday Similarly to the birthday paradox, in which there is a probability that exceeds 0.5 that two people have the same birthday in a room of 23 people. Birthday is a probabilistic protocol, in which two nodes randomly select k slots, where the first listens during its k slots and the second sends a beacon during its k slots. During the remaining time, each node is idle.

Prime-based schemes

Prime number-based asymmetric discovery schemes require a node to choose a single (e.g. U-Connect [37]) or a pair of prime numbers (e.g. Disco [36]) to derive its duty cycle. The activity slots of a node will be the multiples of the selected prime number(s). As an outcome of the Chinese Remainder Theorem, such wake-ups, at multiple of prime numbers, ensures deterministic discovery latency.

Disco In the Disco protocol [36], each device chooses two prime-numbers (p_1, p_2) . Where, Every p_i -th slot the device wakes-up and stay active for one slot and then turns radio-off. Disco sends a beacon at both the beginning and end of each active slot. A mutual discovery between two nodes $((p_1, p_2), (p_3, p_4))$, is assured after $\text{Max}(p_1 \times p_3, p_1 \times p_4, p_2 \times p_3, p_2 \times p_4)$ slots [59].

U-connect As an extension of Disco, the node selects only one prime number p . Thus, U-connect reduces the number of listening periods and beacons. [37] allows for two nodes which choose a prime p to wake-up for a slot every p slots and for $\frac{p+1}{2}$ slots every p^2 slots. The node transmits continuously during listen active slots, thus a node sends $p + \frac{p+1}{2}$ number of beacons. The U-Connect protocol is periodic in nature, with a worst-case latency of p^2 .

Dynamic/Fixed listen slot-based schemes

The transmission and listen activities are independent and they can be conducted on different slots. In dynamic listen slot schemes, a large time period named Hyperperiod T is divided into regular sized cycles $T = c * t$, where each cycle c is further composed of slots. Two types of slots exist, static transmission slots at fixed positions, either at the beginning or end of the cycle, and dynamic listen slots with a regular shift to the left or right in consecutive cycles, up to the end of the period.

Searchlight Under Searchlight scheme [8], a node wakes up for two active slots per cycle c . In every cycle, there is one anchor slot (fixed slot) fixed to the first slot, and a probe slot(dynamic) that traverses from position 1 to $c/2$ across $c/2$ cycles. The number of cycles in a hyperperiod T is $c/2$. In every period, a node sends a beacon at both the beginning and end of an active slot. Under symmetric duty cycles, a node under total active periods and number of beacons sent, are c and $2c$, respectively.

Blinddate Blinddate divides a cycle on m blocks, with only a fraction k of them as dynamic active blocks and one static active block. It places static active slots for deterministic discovery at the final slot of each cycle, then adds dynamic active slot in a pseudo-random manner, moving from left to right in a given dynamic block. Blinddate adopts the same

beaconing strategy as Searchlight. The authors choose the best solution ($k = 2$ and $m = 5$), which results in a 3 active slots and 6 sent beacons per cycle.

Hello For Hello protocol [6], a node stays active during the consecutive slots indexed from 1 to $c/2$ in the first cycle, then wakes up at the first slot of each cycle, which results in $c/2 + t$ listen periods. Similarly to Disco and Searchlight beaconing strategy, a node transmits a beacon at both the beginning and end of an active slot. The number of transmitted beacons per node is $c + 2t$.

Nihao As previous works only care about the overlap of two wake-up slots, Nihao neighbor discovery family uses a strategy named Talk More Listen Less (TMLL), which as its name suggests, it is based on the assumption that when the number of beacons increases, fewer probes are necessary for discovery. TMLL is energy-efficient, given the fact that short beacon costs much less than an active slot. In the S-Nihao variant, a node wakes up in the first slot of each schedule cycle, but sends a beacon at the beginning of each slot. Instead in the G-Nihao variant, a node can skip several slot to send a beacon.

Discussion

For stochastic-based schemes, energy efficiency is ensured by choosing a lower probability for beacon transmission or for listening. Such schemes perform better on the average case compared with the deterministic approaches above, but they provide no bound on the worst case latency and they can lead to long tails in discovering the last fraction of nodes. On the other hand, for Nihao [7], energy efficiency is ensured by choosing to increase beacon transmissions and to decrease listening periods. Such scheme perform better on a small-scale network. The above mentioned schemes are only tested in small scenarios, where two nodes try to discover each other within a bounded delay. On larger scenarios, for a neighborhood size of 100 nodes, we clearly observe that none of the existing schemes enable the nodes to discover all their neighbors [60]. We believe that due to multiple nodes transmitting beacons simultaneously, fraction of nodes fail to discover their neighbors due to collisions.

4.3 WELCOME: Towards a Flock Discovery

4.3.1 System Model

We consider a set of nodes $N = \{n\}$ in the proximity (communication range) one of each other, forming a clique like network structure where each node is with degree $k_n = |N| - 1$. To study neighboring nodes discovery, we assume a relatively stable mobility, i.e. for a while, the nodes stay in the same neighborhood. A node can be either in an active state where it can transmit a message/listen to incoming messages or in a sleep state and remains idle with minimum energy consumption. Thus, a node operating on low duty cycles alternates between sleep and active state where it stays in sleep state most of the time in order to save energy. It becomes active for a small amount of time t_b to transmit a beacon message, or during time t_l to listen to incoming beacons issued by other devices, in a relatively larger time period T , where $t_b < t_l \ll T$. A node employs a given schedule to either send beacons or listen to beacons from another nearby node in the same time period. The goal is to opportunistically find a time when two nodes are simultaneously active to ensure a successful discovery. The energy consumption E_n of the node n to be active as the combination of sending beacon or listening can be represented as $E_n = be_b + le_l$, where b represents the number of transmitted beacons, and e_b is the energy a node takes to transmit a beacon. Similarly, l is the number of listen periods each with energy consumption e_l . The latency for the node n to discover its neighbors is L_n , in the worst case. We define two types of nodes, delegate and member nodes. Delegate nodes are responsible for the neighbor discovery process. They maintain and share the neighborhood information with nearby nodes. Member nodes are the nodes in a particular neighborhood receiving the information regarding their neighbors from delegate nodes.

We define the following basic message types. A delegate node can send two types of messages: (i) A *Discovery* message which marks the beginning of a neighbor discovery process. (ii) A *WELCOME* message which is periodically broadcast in the neighborhood to inform member nodes about current neighborhood information as well as the schedule for the next neighbor discovery process. Any node can send (iii) A unicast *Hello* message to a delegate node in order to provide information regarding its existence in the neighborhood and become a member node.

4.3.2 Delegate Node Eligibility

We believe it is sufficient to allow a single node to declare itself as a delegate node, to represent the neighborhood to which it belongs, in a distributed manner, and welcome incoming neighbors. Nodes in a neighborhood can self-organize and rotate the role of being a delegate among spatio-temporally co-located nodes to ensure a fair energy consumption of all nodes.

We present an eligibility function for a node to become a delegate for some period in a neighborhood based on its residual energy levels. Besides energy requirements, several other factors can be considered depending on the application. Though, one of the important factors is the node reliability in terms of its association to the neighborhood. More precisely, we need to find for how long the node belongs to the same neighborhood in order to avoid that a malicious node, who is new to the neighborhood, with maximum residual energy level to declare itself as the delegate node and subsequently hijacks the neighborhood. Thus, the longer the node is in the same location (i.e part of the same neighborhood), the more likely it is eligible to become the delegate node. Moreover, we need to consider the node history of holding the delegate role, as nodes frequently becoming a delegate are more likely to exhaust their energy earlier compared to their neighbors. Therefore, such nodes should be less preferred to become delegates.

Each node can compute an eligibility function f_D to measure its ability to become a delegate once it becomes active in a neighborhood. It can compute the following function prior to scanning for existing delegates in order to consider itself as the delegate in case of absence of any delegate in the neighborhood:

$$f_D = \alpha f_R + \beta f_E + \gamma f_H \quad (4.1)$$

where, f_R is the node reliability function to be a delegate node taking into account its association (i.e. for how long it belongs) to a neighborhood. The function f_E is the residual energy function characterizing the node physical properties where high node residual energy levels yields high eligibility to become a delegate node. Here f_H is the function which considers the node history of holding the delegate role previously as well as the time

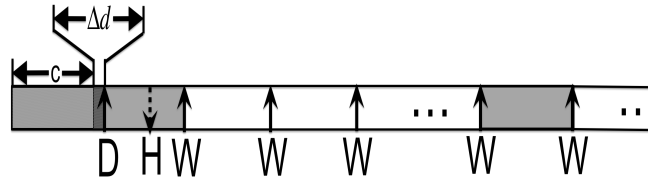
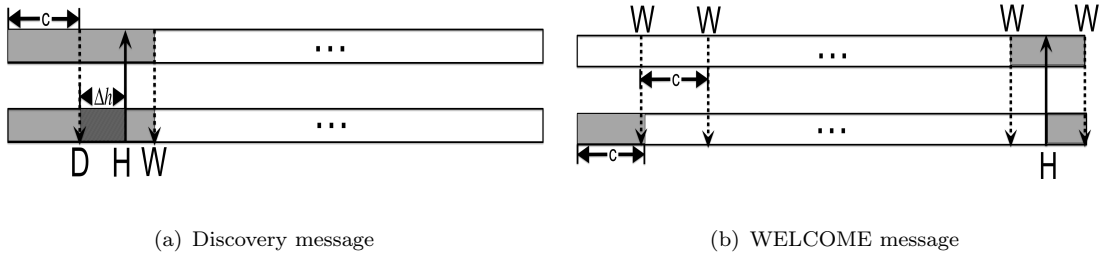


Figure 4.1: Discovery initialization - no delegate node detected



(a) Discovery message

(b) WELCOME message

Figure 4.2: Discovery with existing delegate node message detection

staleness with respect to the previous time it held the delegate role in the neighborhood. An exponential decay function can be used to represent such time staleness. Thus, the more frequent it holds the delegate role since its deployment in the neighborhood, the less it is preferred to be a delegate again, thus increase its lifetime.

The function f_D indicates how eligible a node is to become a delegate, where in case there exists no delegate in the neighborhood i.e. when no WELCOME/Discovery message is received for some time, it waits for an amount of time which is inversely proportional to f_D before declaring itself the delegate and sending its WELCOME message. This not only impedes low residual energy nodes (lower f_E) to become delegates but also reduces the likelihood of a newly joined malicious node (low f_R) or a node frequently selected as delegates in the past (high f_H) to become delegate.

4.3.3 Node activity schedule in WELCOME

Node activity schedule in WELCOME can be classified into two categories: the schedule for the delegate node and the one for member nodes.

4.3.3.1 Delegate Node

Any node becoming active in a neighborhood computes its eligibility score f_D upon wake-up and listen for the existence of a delegate node for a period c as shown in [Figure 4.1](#). In case it receives no message from a delegate, it declares itself as the delegate for the neighborhood by broadcasting a Discovery message after a duration $\Delta d \leq c$. Δd is inversely proportional to its eligibility function f_D with a maximum delay of c period. The node still continues listening for the existence of possible delegate during Δd and aborts transmitting its Discovery message in case it receives a message from an existing delegate node. Subsequent to sending the Discovery message, the node listens for another c duration in order to receive Hello messages from the neighbors that heard its Discovery message.

It is to note that a shorter listen period c can lead to collisions since the growth in the neighborhood size results in more Hello messages sent by neighbors. Therefore, c should be flexible enough for a delegate node to accommodate the reception of Hello messages from all its neighbors while avoiding collisions.

At the end of its listen period, the node then broadcasts a WELCOME message comprising its node ID and the list of neighbors from which it received Hello messages during the listen period. Similarly it contains information regarding the next listen period scheduled by the delegate node based on its duty cycle. The node can then switch to sleep mode and periodically broadcast WELCOME messages to ensure neighboring nodes to detect it. The interval between sending two WELCOME messages is less than the defined listen period c . The idea is to allow the delegate node to switch itself to sleep mode between two WELCOME messages and in result conserve its energy based on its desired duty cycle. Moreover, as shown in the [Figure 4.1](#), the delegate node repeats its discovery phase (listen to Hello messages from neighbors) for the time allowed by its eligibility function and depending on its duty cycle. [Algorithm 1](#) summarizes the neighbor discovery process for a delegate node.

In case the current node doesn't win the election phase and another node outstripped the demand, the candidate node will turn to a Member node mode and execute the following

[Algorithm 2](#). It is to note that the feasibility score calculation and the listening period are only executed upon wake up, otherwise they are escaped.

4.3.3.2 Member Node

A member (non-delegate) node upon arriving in the neighborhood becomes active, computes its eligibility function using [Equation \(4.1\)](#) and listens for a message from a potential delegate node during c period. It can receive either a Discovery message or a WELCOME message from an existing delegate node as shown in [Figure 4.2](#). [Figure 4.2\(a\)](#) shows the case when it receives a Discovery message indicating that the delegate node already started listening for c duration. The member node responds with a uni-cast Hello message to the delegate scheduled after a time delay Δh . This is randomly chosen between the time the member node receives the Discovery message and the time the delegate node finishes its listen period indicated as the duration c . Thus, the size of the interval c is defined as the maximum time to defer a Hello message. The node then continues listening in order to receive the WELCOME message containing information regarding all the neighbors detected by the delegate node. A member node can switch itself to sleep mode following the reception of the WELCOME message and can schedule a wake-up based on its desired duty cycle for an upcoming discovery phase indicated in the delegate's WELCOME message.

The duration of c for a given application should be long enough to accommodate potentially large neighborhood sizes allowing a delegate to be able to listen to Hello messages from a maximum amount of neighbors (member nodes).

The member node upon receiving any of the WELCOME messages sent by the delegate node can defer its transmission of the Hello message to the delegate's upcoming discovery phase as shown in the [Figure 4.2\(b\)](#). Thus, upon reception of a WELCOME message, the member node can switch to sleep mode and schedule to send its Hello message at a time instant randomly chosen between the beginning and the end of the upcoming listen period c , indicated in the delegate's WELCOME message. It subsequently sends a Hello message at the scheduled time and then listens for the WELCOME message following the delegate's discovery phase. Then the member node can return to sleep mode based on its desired duty cycle. The [Algorithm 2](#) shows the neighbor discovery process for a member node.

In the absence of a delegate node, the member node will not receive any messages and will turn to Delegate node mode. As it has already computed its feasibility score and has listened for a c duration upon wake up, these two steps are escaped. Then the current node will listen for an additional time inversely proportional to its eligibility score, thus it enters the Delegate election phase. Nodes that already belong to the neighborhood are favored by the eligibility function f_d and can compete with new arrivals. If one node loses the competition, it turns back to the Member node mode and enters Discovery phase as described above.

Algorithm 1 Delegate Node Activity schedule

```

for node  $n$  do
  Upon wake-up :
  (i)Compute eligibility  $f_d$ , (ii)Listen for  $c$  duration
  if No message received then
    Continue listening for additional  $\Delta d$  period
    Broadcast Discovery message
    Listen for Hello messages for  $c$  duration
    Broadcast WELCOME message
    Sleep and wake-up to broadcast WELCOME message after each  $c$  period
    Periodically do a Discovery phase for  $c$  duration each  $c \cdot t$  slots
  else
    Declare as Member node Algorithm 2
  end if
end for

```

4.3.4 Absence of WELCOME message - Self-organization

A node declaring itself as a delegate is responsible for sharing the neighborhood information for a finite duration allowed by its eligibility function, however, over a longer time period, the role of the delegate is rotated to ensure fairness. A delegate node with low residual energy levels can abdicate itself from being the delegate and subsequently switch to become a member node and stay in sleep mode for longer period in order to conserve energy.

Algorithm 2 Member Node Activity schedule

```

for node  $n$  do
  Upon wake-up:
  (i)Compute eligibility  $f_d$ , (ii)Listen for  $c$  duration
  if Discovery message received then
    Send Hello with waiting time  $\Delta h$ 
    Listen for WELCOME message
  else if WELCOME message received then
    Sleep and schedule a wake-up randomly during delegate's next listen period
    Send Hello message upon wake-up
    Listen for WELCOME message
    Sleep
  else
    Declare as delegate node Algorithm 1
  end if
end for

```

We consider the possibility for a delegate node to stop broadcasting messages due to either its departure from the neighborhood or switching to member node and sleep for energy conservation purposes. WELCOME comprises a built-in recovery mechanism allowing nearby nodes to auto-organize and collaboratively preserve neighborhood information in case of disappearance of messages from the existing delegate node for any reason.

The self-organization process is as follows: Each member node receives and stores the neighborhood information from its delegate node. Once the delegate nodes stops broadcasting its WELCOME message in the neighborhood, similarly to the initial delegate node declaration process, any member node can become the next delegate by broadcasting its respective Discovery message after a wait period Δd that is inversely proportional to its eligibility function f_D in [Equation \(4.1\)](#).

Since our eligibility function considers the nodes history as its association with respect to the neighborhood, only nodes already in the neighborhood for some time in the past will have a shorter waiting period. Moreover, since a node has already stored the neighborhood information from the previous delegate nodes, it can preserve this neighborhood information along with discovering any new nodes joining the neighborhood. Thus, the newly declared delegate node belongs to the same neighborhood while retaining the information regarding the previously known neighbors.

4.4 Analytical Evaluation

4.4.1 Evaluation Metrics

Three evaluation metrics are used in the following. The first metric is the number of discovered neighbors. This one differs between nodes and can be characterized by the average number of neighbors discovered in a neighborhood as defined below:

Definition 4.1. Fraction of Discovered neighbors

The number of neighbors discovered by a node n is defined as the cardinality of $D_n \subset N$, the set of neighbors discovered by n . Similarly, the average number of neighbors discovered for a set N is the cardinality of the set D_N represented as $D_N = \frac{1}{|N|} \sum_{n \in N} D_n$, where the unit of both D_n and D_N is a number of nodes ranging between 0 and $|N| - 1$.

The worst case latency in discovering neighbors is an important metric, used by all related works in [Chapter 2](#). However, in the case where only a fraction of neighbors is discovered, we need to consider the joint relation between latency and average neighbor discovery. This relation, which represents our second evaluation metric is defined here after.

Definition 4.2. Latency vs. Discovery

The latency vs discovery relation for a neighbor discovery process considering a possibility of failure in discovering neighbors for a set of nodes $|N|$ is given as:

$$\theta_N = L_N \cdot \left(1 + \frac{D_N}{|N| - 1}\right), \quad (4.2)$$

where L_N can be seen as the theoretical worst case latency, when ignoring collisions, for discovering the $|N| - 1$ neighbors, and D_N is the average number of neighbors currently discovered. The term θ_N can be interpreted as a measure of a general performance (i.e. the average number of time slots needed to discover a certain ratio of nodes) and a relative one (i.e. the average number of neighbors discovered in practice during a time period where all the nodes should have been theoretically discovered). It can be measured in term of time slots.

We also need to consider a measure on the energy efficiency, as the discovery process should be as less energy consuming as possible, while incurring low latency and providing a high discovery ratio. Therefore, we jointly characterize the energy consumption, the average number of neighbors discovered and the average latency, using the relations below.

Definition 4.3. Energy vs. Discovery

We define, for a set of nodes N , the relation between the total fraction of neighbors discovered, the energy consumption and the latency needed for discovery as:

$$\delta_N = E_N \cdot \theta_N, \quad (4.3)$$

where E_N is the average energy consumption during time period L_N . The metric δ_N is measured in Joule-second and it provides a common benchmark for the different neighbor discovery schemes.

4.5 Theoretical Comparison

4.5.1 key parameter

Duty Cycle

The listen period consumes most of the node energy and almost all nodes listen for a duration $c \leq c + \Delta \leq 2c$ and sends one beacon during this same active period $c + \Delta + 1 \approx c + \Delta d$. Except for the Delegate node, that listens for a duration $2c \leq 2c + \Delta d \leq 3c$, and every c slots, it wakes-up to send one beacon. Thus the active period for Delegate is $2c + \Delta d + t$ slots. The duty cycles corresponding to Member node and Delegate node are $DC_m = (c + \Delta)/ct$ and $DC_d = (2c + \delta d + t)/ct$, respectively. The parameter t can be defined by an application as the number of times the listen period c is to be repeated, reflecting the frequency of the re-initialization of the discovery process by the delegate node. Therefore using our proposed protocol, we consider the duty cycle of the Member node as Welcome duty-cycle.

Worst-case Latency

For the delegate node, the worst case latency is $L_n = 2c + \Delta d$ since after waiting this amount of time, it is supposed to receive all the Hello messages and then broadcast its WELCOME message.

The worst case latency for the member node receiving a Discovery message is $L_n = c + \Delta$ as it discovers its neighbors from the information in the WELCOME message it receives following the Discovery message from the delegate node. The worst case latency for a member node for receiving a WELCOME message is the discovery cycle of the delegate node, i.e. $L_n = c \times t$. Thus, for a member node, the worst case latency of $L_n = c \times t$ is considered a maximum bound under the WELCOME protocol.

Energy Consumption

The energy consumption of the delegate node is given as $E_n = (2c + \Delta d)e_l + te_b$ where the number of beacons are $t + 1$ and the number of listen periods are $2c + \Delta d$. For the member node, the energy consumption is $E_n = (c + \Delta)e_l + e_b$, where at worse, it listens for a duration of $c + \Delta$, and sends a single Hello message at the Delegate next listen period. When a member node wakes-up at the Delegate node listen period, it can face one of the two possible scenarios. At best, it listens for c slots and get one of the possible messages (Discovery/Welcome message), in this case the listening period is c slot and the number of sent beacons is one. At worst case, the member node doesn't receive any message, then it switches into Delegate node mode, waits for an additional Δd time, then turns back into Member node mode.

4.5.2 Detailed theoretical analysis

Similar to WELCOME, we theoretically derive the key parameters for each of the best state of the art schemes such that all nodes achieve the same desired duty cycle and ensure fair comparison. Though such schemes can function under heterogeneous duty cycle settings and this step is only required for comparison purposes. Specifically, we are interested in the nodes activity schedule, their energy consumption on transmission and/or listen time slots, as well as the latency for successful neighbor discovery when operating on a

Table 4.1: WELCOME vs Existing Neighbor Discovery Schemes (Key parameters)

Scheme	Parameter(s)	Duty Cycle	Beacons	Listen periods	L_n	E_n
WELCOME [Delegate]	$c, t, \Delta d$	$(2c + \Delta d + t)/ct$	t	$2c + \Delta d$	$2c + \Delta d$	$(2c + \Delta d)e_l + te_b$
WELCOME [Member]	c, t, Δ	$(c + \Delta)/ct$	1	$c + \Delta$	ct	$(c + \Delta d)e_l + e_b$
Hello [6]	c, t	$(c/2 + t)/ct$	$c + 2t$	$c/2 + t$	$c^2/2$	$c(2e_b + e_l)$
G-Nihao [7]	c, t	$c + 1/ct$	t	c	ct	$c(2e_b + e_l)$
Disco [36]	p_1, p_2	$1/p_1 + 1/p_2$	$2(p_1 + p_2)$	$p_1 + p_2$	$p_1 p_2$	$(p_1 + p_2)(2e_b + e_l)$
U-Connect [37]	p	$3/(2p)$	$3p/2$	$3p/2$	p^2	$3p(e_b + e_l)/2$
Searchlight [8]	c	$2/c$	$2c$	c	$c^2/2$	$c(2e_b + e_l)$
Blinddate [9]	s, t	$3/(5s)$	$6t$	$3t$	$5s^2/2$	$3t(2e_b + e_l)$
Birthday [10]	p_b, p_l, p_s	$p_b + p_l$	$p_b L_n$	$p_l L_n$	-	$L_n(p_b e_b + p_l e_l)$

Table 4.2: WELCOME vs Existing Neighbor Discovery Schemes (Results)

Scheme	1% duty cycle						5% duty cycle					
	L_n	E_n	c	t	s	$p_1, p_2/p$	L_n	E_n	c	t	s	$p_1, p_2/p$
WELCOME [Delegate]	300	$200e_b + 300e_l$	100	200			60	$40e_b + 60e_l$	20	40		
WELCOME [Member]	20000	$e_b + 200e_l$	100	200			800	$e_b + 40e_l$	40	20		
Hello [6]	20000	$400e_b + 200e_l$	200	100			800	$80e_b + 40e_l$	40	20		
G-Nihao [7]	20000	$400e_b + 200e_l$	200	100			800	$40e_b + 80e_l$	40	20		
Disco [36]	40301	$804e_b + 402e_l$		211		191, 211	1591	$160e_b + 80e_l$		43		37, 43
U-Connect [37]	22801	$228e_b + 228e_l$		151		151	961	$48e_b + 48e_l$		31		31
Searchlight [8]	20000	$400e_b + 200e_l$	200	100			800	$80e_b + 40e_l$	40	20		
Blinddate [9]	9000	$180e_b + 90e_l$	300	30	60		360	$36e_b + 18e_l$	60	6	12	
Birthday [10]	20000	$200e_b + 200e_l$				200	800	$40e_b + 40e_l$				40

particular duty cycle. Table 4.1 summarizes this comparison for nine different neighbor discovery schemes, schemes that are discussed in Chapter 2. In the following, we limited the theoretical bounds comparison for Slotted Protocols, as the fundamental performance bound for periodic-interval(PI)-based protocols (i.e. [4],[5]) are still unknown [61]. The first column in Table 4.1 shows the key parameters used to define a node activity schedule in order to attain a given duty cycle, using the relation shown in the second column. The subsequent columns show the number of beacons, listen periods, worst case latency bound L_n and energy consumption E_n , respectively, for each scheme.

In the following theoretical comparison, given by Table 4.2, we provide the numerical values of the worst case latency and energy consumption of the same set of Neighbor Discovery schemes while comparing them to WELCOME. More precisely, Table 4.2 gives an outline of the derived numerical values of E_n and L_n , for a node operating at 1% and 5% duty cycles, respectively.

Giving a duty-cycle (e.g. 1%, 5%), the optimum c value can be achieved based on what is expected in the second column of [Table 4.1](#). To compare G-Nihao to other schemes, we use a sufficiently small $\alpha \leq Dc$ to ensure a fine-grained duty-cycle in the form of $t = \frac{c}{Dc}$ [7]. The best cycle length c configuration for both Searchlight and Hello, is $2/Dc$. Indeed, once we fixed one parameter c in function of Dc , it is easy to calculate the number of cycles t . For Disco, we set balanced primes close to $\frac{2}{Dc}$, under symmetric duty cycles. U-Connect fixes its prime p close to $\frac{3/2}{Dc}$. Concerning Blinddate, the optimal total number of time slots in a block is $s = 3/(5Dc)$. Subsequently, the cycle length is 5 times the block size, so $c = 5s$. From the worst case latency equations $Ln = 5s^2/2 = ct$ [9], we can derive $t = Ln/c$. Due to the stochastic nature of Birthday there is no bound on the worst case latency for discovery, thus, in order to ensure a fair comparison with other benchmark deterministic schemes, we consider a worst case latency $Ln = \frac{p^2}{2}$, as for Searchlight, Hello, and for Nihao. Due to the asynchronous nature of WELCOME, as the Delegate and Member nodes operates under different duty-cycles, the optimum cycle length is set to $1/Dc$ [6].

We compare the performance of the different protocols under symmetric duty cycles. Two typical duty cycles, namely 1% and 5%, are used hereafter. According to the parameter selection rules described above, each scheme selects its optimal parameter values, look at the column 1 in the table reference [Table 4.1](#), under both duty-cycles. Disco selected (191, 211) and (37, 43) (balanced primes), U-Connect 151 and 31, both G-Nihao and Searchlight 200 and 40, Hello (199, 100) and (41, 19), Blinddate (300, 60, 30) and (60, 18, 6), Birthday (200, 200). For Welcome, when Member nodes operates under 1%/5%, the Delegate node for the same cycle length operates under 2, 5%/12, 5%, and the optimum cycle length and number of cycles are 100 and 200, respectively.

The energy consumption of Disco under 1% duty-cycle, is $En = 804eb + 402el$, i.e. $2(p_1 + p_2) = 804$ beacons and $p_1 + p_2 = 402$ listen periods within the bounded delay $Ln = 40301$ time slots. Similarly, to achieve 5% duty cycle within $Ln = 1591$, the energy consumed by a node is $En = 160eb + 80el$ with 80 active time-slots. To achieve 1% duty cycle, U-Connect requires an energy consumption of $En = 228eb + 228el$ within the worst case latency $Ln = 22801$ slots, where the number of beacons and listen periods are both 228. For 5% duty cycle, within the worst case discovery period of 961 slots, a node spends energy to send 48 beacons and listens for 48 slots. Searchlight and Hello, under 1% duty

cycle, the number of beacons and listen periods are 400 and 200, respectively, leading to an energy consumption $En = 400eb + 200el$ within the worst case latency $Ln = 20000$ slots. For 5% duty cycle, The energy consumption is $En = 80eb + 40el$ within the worst case latency Ln of 800 slots.

Blinddate requires a cycle of 300 time slots repeated $t = 30$ times, under 1% duty cycle. The cycle is divided into 5 sub-slots each of size $s = 60$ slots with a worst case latency of 9000 slots. The number of active periods is 90, thus a node send 90 beacons and spend 90 listen periods and the energy consumption is $En = 90(2eb + el)$. Similarly, under 5% duty cycle, Blinddate requires a cycle of 60 repeated $t = 6$ times with a worst case latency of 360 slots. The cycle is divided into 5 sub-slots each of size $s = 12$ slots. The number of beacons and listen periods are 18 each leading to an energy consumption $En = 18(eb + el)$. (4.3)

For 1% duty cycle, G-Nihao needs up to $Ln = 20000$ as the worst case delay. The energy consumption in this case is $En = 200(2eb + el)$. Similarly, for 5% duty cycle, the worst case delay is $Ln = 800$ slots with an energy consumption of $En = 40(2eb + el)$. (4.4)

Finally, WELCOME requires a cycle period of $c = 100$ slots repeated $t = 200$ times, to achieve 1% duty cycle. For a Delegate node, it sends 200 beacons and listen for 300 periods resulting in an energy consumption of $En = 200eb + 300el$. Similarly, for a Member node, it sends 1 beacons and listen for 40 slots resulting in an energy consumption of $En = eb + 40el$. The worst case latencies, for a Delegate Node and Member Node are $Ln = 300$ and $Ln = 20000$, respectively. Under 5% duty cycle, a Delegate Node energy consumption is given as $En = 40eb + 60el$, where $2c + \Delta = 60$ is the number of listen periods and $t = 40$ is the number of listen slots for the WELCOME schedule within the bounded delay.

4.5.2.1 Discussion

From the evaluation above, we can conclude that Blinddate and WELCOME reach the best theoretical performances, better than all the other schemes. More specifically Blinddate reaches the best performance in terms of worst case latency, while WELCOME reaches the best theoretical performances in terms of energy consumption. On the other hand, Disco results in the worst performances among the compared schemes. In terms of both

energy consumption and latency, Disco requires twice the other compared schemes expect for Blinddate, where it costs even four times more. The other protocols (WELCOME, Searchlight, Hello, Birthday, G-Nihao and U-Connect) reach a relatively equal worst case latency results. Concerning the Energy consumption, the best performance is reached by WELCOME, which saves per node from 200% up to 800% in terms of beaconing dedicated energy. In terms of listening dedicated energy, it consumes the same amount of energy as Hello, G-Nihao, Searchlight and Birthday. Blinddate, however, consumes twice less listening dedicated energy, while Disco consumes two times more listening dedicated energy than WELCOME. On the other side, the Delegate node has an average energy consumption compared to other schemes, however it represents 1% from the network and in addition this charge is fairly distributed in time over the network nodes.

The remarkable difference in the total number of beacons sent by WELCOME compared to other schemes supports more the assumption that WELCOME scales far better. Since large networks are more prone to collisions, we believe that WELCOME is able to reduce the worst-case (long tail) latency significantly, and could make difference on (near) realistic scenarios. Besides this encouraging theoretical comparison, it is still necessary to analyze the performances of WELCOME under realistic simulation scenarios. This is the target of the following section.

4.6 Simulation-based Evaluation

4.6.1 Simulation Scenario

Simulations are performed by implementing WELCOME along state of the art discussed neighbor discovery mechanisms in the NS-3 simulator. A neighborhood is formed by placing a set of 100 IEEE 802.11b/g/n enabled nodes in the communication range (around 150 m) of each other in an opportunistic and self-organized network, resulting in a clique-type network. We assume nodes stay in the neighborhood for some time. i.e. the topology does not evolve during our analysis. We evaluate each mechanism by considering up to 100 nodes, where the Friis propagation loss model is used to study the impact of fading in the wireless medium. Since energy efficiency can be achieved by allowing a node to operate on low duty cycles, we consider low duty cycles of 1% and 5%. We vary the number of co-located nodes from 2 nodes up to a total of 100 nodes simultaneously present

in a neighborhood. This encompasses not only few mobile nodes discovering each other for applications such as mobile sensing or proximity-based gaming but also allow large number of energy constrained sensors on shipment packages to discover each other autonomously for tracking purposes for instance.

Each member node sends a Hello message of 100 bytes while a delegate node sends a Discovery message and WELCOME message of 100 bytes and 1KB, respectively. Similarly, a node can listen during a duration of $c = 200$ ms, which is large enough to accommodate Hello messages from up to 100 nodes. The parameters α , β and γ in Equation (4.1) are set to 0.33 to maintain generality for each function. We consider the possibility of a clock drift between nodes as an asynchronous discovery since they are unaware of the time lag between each others active periods. WELCOME is compared to the best among the above mentioned schemes *i)* Blend and *ii)* Garissidi, *iii)* fixed-slot based Hello, and *iv)* G-Nihao, *v)* dynamic slot based Searchlight, *vi)* Blinddate, and *vii)* stochastic Birthday, under symmetric duty-cycles. Each node follows an activity schedule using the respective parameters defined by each mechanism in Table 4.1. Since each of the deterministic mechanisms discussed above theoretically ensures a successful discovery if the neighboring nodes are active during L_n , the simulation duration ensures that each node experiences at least L_n , as the time period in common with all its neighbors.

We recall that each of the above mentioned mechanisms tries finding an overlapping active time period between nodes in order to ensure a successful discovery. However, in practice, activating multiple nodes at the same time can lead to collisions, thus resulting in discovery failures. We tackle the issue by implementing a CSMA/CA based back-off approach where a node finding the medium busy before transmitting a beacon chooses a wait time randomly between its initial transmission time and a slot size of 10 ms.

We use our proposed evaluation metrics: *i)* The average number of discovered neighbors D_N among $|N|$ nodes in each other communication range. *ii)* The Latency vs discovery relation in Equation (4.2) to find the discovery latency incurred by the nodes when applying the schedule defined by each scheme, and *iii)* The Energy vs discovery (δ_N), defined in Equation (4.3), to find the energy consumption of the node using WELCOME and other neighbor discovery schemes.

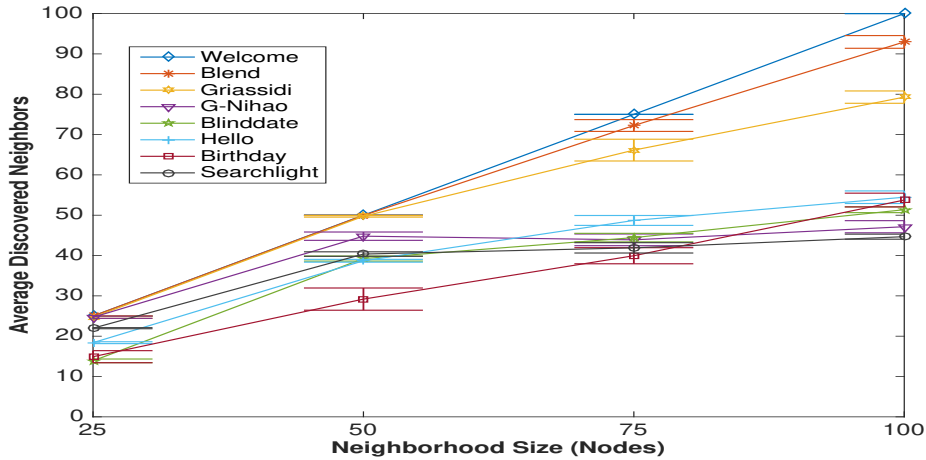


Figure 4.3: Average discovered neighbors comparison

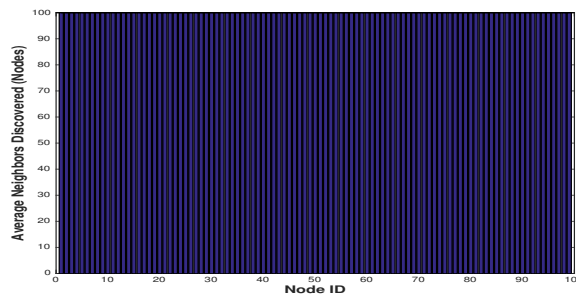
4.6.2 Simulation Results

We performed simulations on nodes running different duty cycles 1% and 5%, respectively. In the following, we present only the results from simulations since using a 5% duty cycle nodes are more prone to collisions. The results are obtained using ten simulation runs, where the average values of the results are shown with 95% confidence intervals.

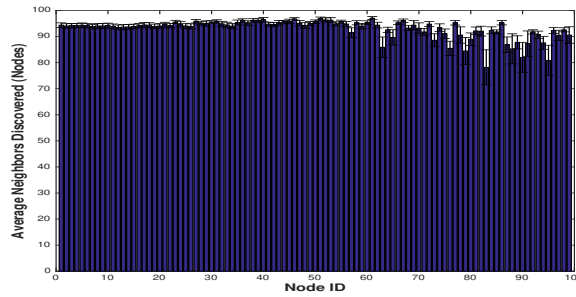
4.6.2.1 Average discovered neighbors

The motivation for proposing WELCOME is the risk of collision between beacons transmitting at the same instant, thus resulting in all the neighbors not necessarily discovered by the node. We investigate such a behavior by finding the average number of neighbors discovered while having different neighborhood sizes. Figure 4.3 compares the average discovered neighbors using WELCOME along different state of the art schemes. It clearly shows that WELCOME yields the highest number of discovered nodes, followed by the indirect schemes Blend and Griassdi. On the other hand, direct neighbor discovery schemes result in the least number of neighbors discovered on average.

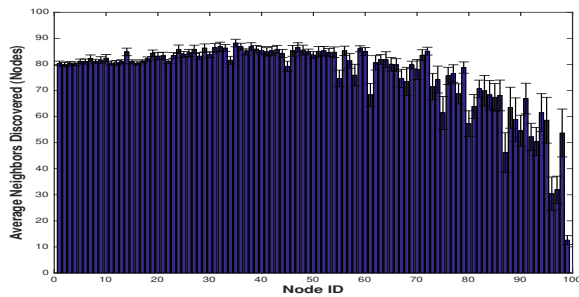
WELCOME discovers 100% of the neighbors irrespective of the neighborhood size, and therefore, outperforms other schemes where Blend discovers around 90% of the nodes. One possible reason for WELCOME high performance is the fact that it avoids collisions at the active overlapping time for discovery allowing a single node to listen to Hello messages from neighbors, which significantly increases the chances to discover neighbors.



(a) WELCOME



(b) Blend



(c) Griassdi

Figure 4.4: Histogram of average discovered neighbors

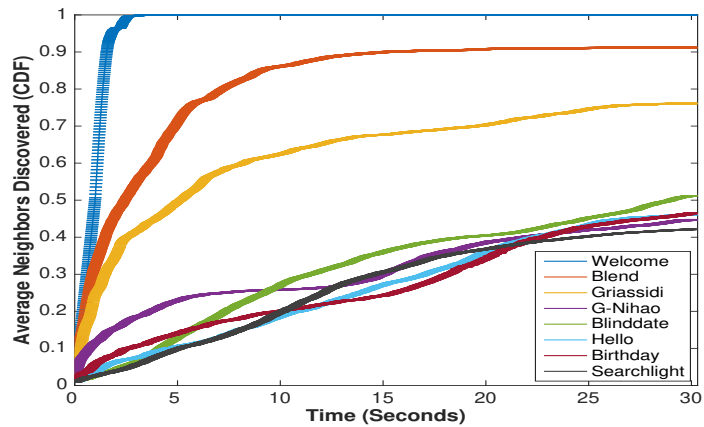


Figure 4.5: CDF comparison of average discovered neighbors

We found that among all the compared schemes, Searchlight resulted in the poor performance because the low listen period reduces its chance of discovering a large fraction of nodes as we can observe that less than 50% of nodes are discovered for a neighborhood size of 100 nodes.

Furthermore, we analyzed the scalability of WELCOME by comparing the neighbors discovered by individual nodes in the large evaluated neighborhood of 100 nodes. [Figure 4.4](#) shows the histogram of the neighbor discovered by each node compared with the best performing indirect schemes, Blend and Griassdi. It clearly shows that WELCOME (cf. [Figure 4.4\(a\)](#)) allows all nodes to discover their neighbors where the neighborhood information is shared by the delegate node. On the other hand, for Blend and Griassdi in [Figure 4.4\(b\)](#) and [Figure 4.4\(c\)](#), a fraction of nodes failed to discover all the neighbors in the neighborhood size of 100.

Additionally to observe the fraction of neighbors discovered over time, we compute the CDF of the neighbors discovered using each scheme for the neighborhood size of 100 nodes. We observe such phenomena in [Figure 4.5](#) where we compare WELCOME with the state of the art schemes shown earlier in the [Figure 2.1](#) for up to 30 seconds. WELCOME discovered all the 100 neighbors with a delay of less than 5 seconds compared to existing neighbor discovery schemes. It is followed by indirect schemes Blend and Griassdi, then the direct schemes when yielded around similar performance by discovering around 50 neighbors at the delay of around 30 seconds.

Overall the results for average discovered neighbors validate our claims that discovery failures can occur largely due to the collisions of messages transmitted by multiple nodes in a neighborhood. WELCOME tackles such issues by allowing one delegate node to discover neighbors and share this information in a distributed manner. Thus, it results in a scalable low latency, and energy efficient discovery of up to 100% of neighbors for low duty cycle nodes.

4.6.2.2 Latency vs Discovery

We analyze the latency achieved by the nodes using WELCOME when discovering neighbors compared to other state of the art schemes. Figure 4.6 shows the latency vs discovery metric for each scheme, running on nodes with 5% duty cycle. A lower value of the latency vs discovery metric reflects a better performance in terms of latency. We notice that WELCOME results in the best performance, i.e. quick discovery of a high fraction of neighbors in both small-scale and large-scale neighborhood, and thereby validating the scalability of WELCOME compared to both, existing direct and indirect neighbor discovery schemes.

For the state of the art schemes, an increase in latency vs discovery is observed with the increase in neighborhood size, particularly when we compare for large scale neighborhoods (i.e. > 25 nodes). This is because, with the increase in the number of nodes, the chances of collisions of beacons increases as multiple nodes try to transmit simultaneously. WELCOME overcomes this issue by allowing a single node transmitting in the neighborhood at a time and therefore avoiding collisions among beacon messages.

Overall, the comparison of WELCOME with state of the art schemes for the latency vs discovery metric suggests that WELCOME can successfully discover a large fraction of neighbors with the least latency. We can also infer that WELCOME scales better and is relatively stable with respect to the increase in the number of neighbors as well as it remains unaffected by the increase or decrease in the neighborhood size.

4.6.2.3 Energy vs Discovery

We finally analyze WELCOME for the energy vs discovery metric in comparison with other neighbor discovery schemes where a low energy vs discovery value for a scheme correspond to a better performance. Figure 4.7 shows such comparison with respect to this metric when using each neighbor discovery scheme for nodes operating on 5% duty cycles.

It can be clearly shown that WELCOME outperforms all existing neighbor discovery schemes with the least energy consumption for the fraction of discovered neighbors in each neighborhood. We also observe that the indirect schemes, Blend and Griassdi, performs

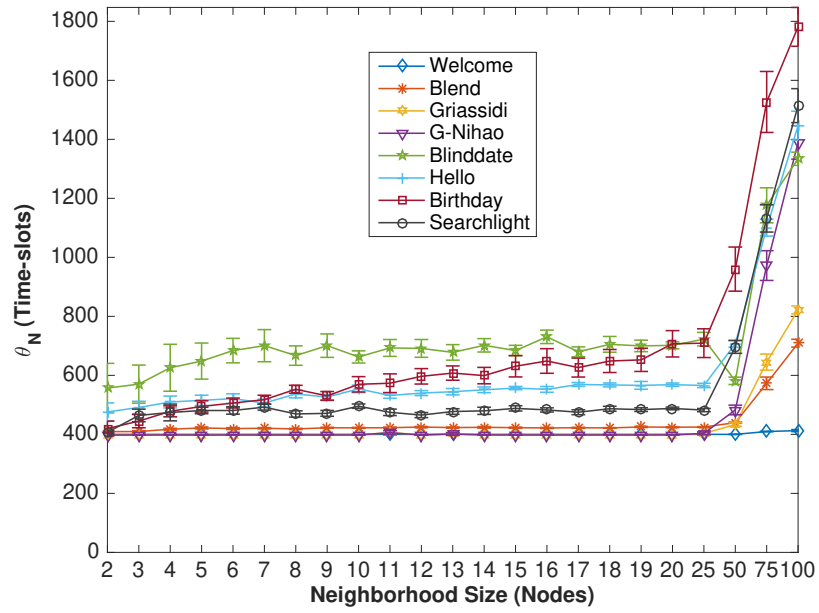


Figure 4.6: Latency vs discovery

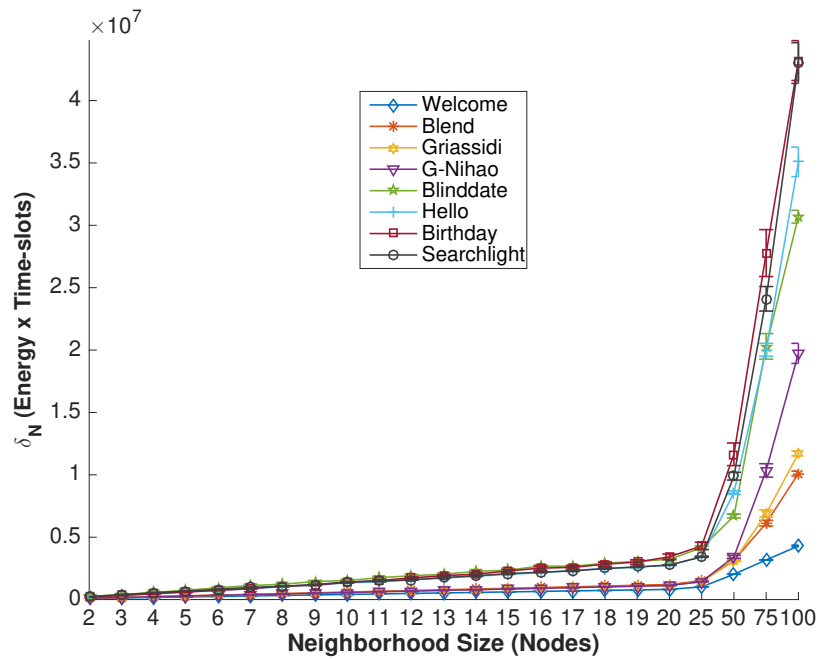


Figure 4.7: Energy vs discovery

better compared to the direct schemes as nodes assist each other to enhance the neighbor discovery process. Still such schemes are consuming significantly high energy compared to our proposed WELCOME scheme. Direct neighbor discovery schemes on the other hand resulted in a the poorest performance, which is the more visible for the probabilistic Birthday, consuming the maximum energy.

Thus, the overall energy vs discovery analysis using simulations suggests that WELCOME is an efficient and scalable neighbor discovery scheme with relatively less energy consumption and maximum amount of discovered neighbors compared to existing schemes in the literature.

4.7 Conclusions

Neighbor discovery for low power devices usually suffer from the large number of collisions between messages transmitted by multiple nodes as the neighborhood size grows. To tackle this issue, we propose a solution where one node at a time can fully discover its neighbors and inform about nearby nodes. Therefore, in this work, we propose WELCOME, a new neighbor discovery scheme where one node can declare itself as a delegate for the neighbor discovery process. WELCOME allows only nodes eligible with respect to their residual energy and neighborhood association to become delegates, thus a node with sufficient energy can become the delegate in case of absence of existing delegate node. WELCOME is evaluated for the amount of neighbor discovered, energy consumption and latency in the discovery process. This evaluation, using simulations, make use of neighborhood sizes of up-to 100 nodes. Results show that WELCOME not only discovered 100% neighbors but also yielded low energy consumption and discovery latency due to the reduction in the amount of transmitted messages leading to collisions.

In the previous chapter, we targeted to ensure the spatio-temporal extraction of IoT devices' context information. We propose a CRF-based model based on a collaboration strategy between available nodes which allows avoiding the deployment of dedicated physicals source. It reaches up to 98.5% accuracy when there is enough close neighbors. However, IoT devices are connected only temporarily and the network topology may change due to node mobility or node activation and node deactivation, respectively. Furthermore, IoT

devices could be battery-powered and can opt for duty cycling. Consequently, the performance of such a model depend firstly on an efficient neighbor discovery. For this purpose we presented in this chapter WELCOME, a low latency and energy efficient neighbor discovery scheme. Once, energy-constrained nodes can discover each other with lowest energy wasting, it is now challenging for current network infrastructure to ensure a context-aware data collection. Selecting the appropriate IoT devices based on their context among the thousands of connected devices on urban areas, is a difficult task, solving this issue is the aim of the next chapter.

Chapter 5

Context-aware and Energy Efficient Publish-Subscribe for IOT: CEEPS4IOT

Contents

5.1	Introduction	76
5.2	CEEPS4 IoT	78
5.2.1	System Model	78
5.2.2	Deriving Object Utility	79
5.2.3	Coalitional Game formation	82
5.3	Performance Evaluation	86
5.3.1	NS3 Simulation-based evaluation	87
5.3.2	Evaluation Results	89
5.4	Conclusions	92

5.1 Introduction

Electronic devices and sensor networks are battery-powered and have severe energy constraints which make the design of energy-efficient message passing protocol a key task. In addition, data collection in Internet of Things can suffer from redundant or unwanted data

transmissions which result mainly in power waste and additional network overhead and latency [44]. From the literature, many energy-efficient and context-aware message passing protocols are proposed. They are based on data filtering, such that only the nodes providing information of interest are selected. However, they focus mainly on location and time-stamp as context features, while context can be a broader notion than simple location and time-stamp information. In this direction, filtering-out data close to data sources based on "T2T Context-awareness at the Edge" could ensure more accurate data collection and at the same time avoid useless information transmissions.

The service provider has to be aware of service's context involved in the application process and especially the ones collecting data. However, such knowledge (context) extraction could require additional data transmissions from neighbors as part of a collaborative approach [1], which may potentially cancel the efforts to save energy with the filtering process based on the "Context at the edge" concept. An optimization solution is needed in order to find, for each node, an optimal trade-off between maximizing the number of neighbors involved in its context estimation phase and, at the same time, minimizing the involved costs in terms of energy saving and network overhead. The arising question to answer in this chapter is this: How to optimally incentivize the best set of neighbors to share data for context discovery task while dealing with IoT devices individual rationality to save their resources?

To answer this question, we propose, in this chapter, **CEEPS4IoT as an energy efficient and context aware Publish-Subscribe system** for IoT that takes into account the context of neighboring sensors while collecting data. Rational sensors do not cooperate since sharing their readings is an energy costly operation along incurring them additional communication overhead. To tackle this, CEEPS4IoT is based on a **Dynamic Coalitional Game** for sensors to collaborate and share their readings in an energy-efficient way and in return receive a reward for cooperating. We derive a stable utility for a sensor proportional to the amount of data it shares while compensating for its energy costs. Results from evaluating CEEPS4IoT in networks of upto 300 nodes suggest it as a scalable and energy efficient location-based Publish-Subscribe system with context awareness since it conserves around 50% energy compared to existing location-based Publish-Subscribe system.

The remaining of the chapter is organized as follows, the next [Section 5.2](#) provide the CEEPS4IoT model where we first derive the device context followed by the coalitional game ensuring an energy efficient data collection from the devices providing high quality data. The performance evaluation and results are discussed in [Section 5.3](#) followed by the conclusions and future directions in [Section 5.4](#).

5.2 CEEPS4 IoT

In this section, we formally describe the proposed Publish-Subscribe system for IoT using a Dynamic Non-Transferable-Utility *NTU*-coalitional game. First, we define below the system model to consider through the rest of the chapter along deriving the context and utility of the devices in coalition. Then, we discuss the coalition game, its stability followed the message exchange required for the energy efficient data collection.

5.2.1 System Model

We define the context of a publisher at time-slot \bar{t} and location l as the physical and/or logical state in which it (sensor/actuator) acts such as being under water/shade. For a publisher, the environment it is installed on can influence and consequently affect the quality of service it provides. We leverage the readings from other coexisting publishers at the same location in order to extract a publisher's context. However, collecting readings from all sensors can lead to additional messages transmitted by the publishers, thereby resulting in an extra overhead and energy consumption. We propose an energy-efficient Publish-Subscribe system to keep track of publishers' contexts where a context manager component or a hub/broker arranges publishers into a set of coalitions of nearby publishers in the same location l and time duration \bar{t} to cooperate in order to obtain a reward from a service/application provider. In the remainder of the chapter, we use device, sensor and publisher interchangeably.

We formulate a dynamic coalition game in a partition form with non-transferable utility by a pair (\mathbb{X}, Φ) , where $\mathbb{X} = \{x\}$ is the set of players (i.e. publishers), and Φ is a mapping function representing the set of coalitions as $\Phi(\mathbb{X}) = \mathbb{S} = \{S_1, S_2 \dots S_k\}$. We define $U : \mathbb{X} \times |\mathbb{S}| \rightarrow \mathbb{R}$, the real-valued function that associates for a device x to join a coalition of

Table 5.1: List of Notations

Notation	Description
$T = \{\bar{t}\}$	Set of time-slots
$L = \{l\}$	Set of locations
$\mathbb{X} = \{x_1, x_2, \dots, x_n\}$	Set of devices/publishers
$\Phi(\mathbb{X}),$	Partition function
$\mathbb{S} = \{S_1, S_2, \dots, S_k\}$	Collection of Coalitions
Γ, Γ'	Partitions of \mathbb{X}
$\Theta = \{\theta\}$	Set of context
$E_{x_i}^S$	Cost function of device x_i
γ	tuning parameter
R_e	Residual energy (Joules)
τ	Time interval between two readings
$G_{x_i}^S$	Gain function of publisher x_i
Δ	Inter device distances vector
d_{ij}	Distance between publisher x_i and x_j
$CMI_{x_i, S_a}(x_i; x_1, x_2, \dots, x_{na}, \Delta)$	Weighted conditional multivariate MI
$MI_S(x_1, x_2, \dots, x_{na}, \Delta)$	Weighted multivariate MI
$U_{x_i}(S_i, \Omega)$	Utility function of device x_i
Ω	Publishers order to enter a coalition S_i
$U_{S_i}^v$	Utility function of coalition S_i at iteration v
i, j	Index of publishers
a, b	Index of coalitions
\succ	Preference relation based Pareto Order
σ	Utility threshold

devices S in a location and during a time-slot (l, \bar{t}) , the utility function U_{x_i} as a reward for collaborating, at time-slot \bar{t} .

5.2.2 Deriving Object Utility

It is challenging for different devices to collaborate given their individual rationality of best preserving their resources and at the same time maximizing the amount of information about their own context. To tackle this, we design the utility function such that the gains for a publisher are proportional to the amount of dataset of nearby devices share about their contexts. Similarly, we derive the costs for a publisher that take into account not only its residual energy but also the amount of data exchanged with its neighboring devices. Thereby, once a device enters a coalition that maximizes its profit, it shares in return its collected data with its neighbors inside the same coalition. The trade-off for a publisher to

join a coalition is a situational decision that involves diminishing its energy and maximizing the information about its context. The utility or payoff for an device in a coalition can be derived from the difference between its respective gains and cost, therefore, we present below the gains and costs for the devices forming coalitions.

5.2.2.1 Gain function

We incentivize each publisher to share information regarding the local environment with its nearby publishers, thus, its gain is modeled such that it is proportional to the information nearby sensors have on its context. The gain for a publisher x_i to join a coalition $\mathbb{S} = \{x_1, x_2, \dots, x_k\}$ is the amount of information the set of near-by publishers CMI_{x_i} have regarding the context of the corresponding publisher x_i , weighted by their respective distances, i.e. the weighted conditional mutual information given as:

$$G_{x_i}^S = CMI_{x_i}(x_i; S, \Delta) \quad (5.1)$$

Δ is the inter publisher distances vector between the device and its neighbors. The gain can be expanded as follows:

$$CMI_{x_i}(x_i; S, \Delta) = CMI_{x_i}(x_i; x_1)(1 - d_{i1})^{-\lambda} + \sum_{j=2}^k CMI_{x_i}(x_i; x_j | x_1, \dots, x_{j-1})(1 - d_{ij})^{-\lambda} \quad (5.2)$$

where, d_{i1} is the distance between the device x_i and the neighboring device s_1 . The term λ is the tuning factor to measure the impact of neighboring publisher's reading on the device x_i context estimation.

5.2.2.2 Cost function

Similarly, we derive the cost for a sensor x_i to join the coalition $S = \{x_1, x_2, \dots, x_k\}$. Different factors can be considered for the device cost in joining a coalition depending on the application requirement, however, in our case, the sensor/device energy consumption is considered as a key factor towards the cost computation in its utility function. The cost E_{x_i} for the sensor x_i can be defined as an exponential decay function where the cost of a

sensor participation in a coalition decreases exponentially to its residual energy R_e and the average number of packets exchanges in the coalition at a location l and during a time-slot t :

$$E_{x_i}^S = e^{R_e - [\gamma \frac{t}{\tau}]} \quad (5.3)$$

Here, γ is the tuning parameter to adjust the cost factor while τ is the time interval between two consecutive readings from the sensor. Thus, for a sensor x_i , the higher number of packets that are exchanged and the higher residual energy directly affects its cost function.

5.2.2.3 Utility function

Given the gains and costs, we can now define the marginal utility U_{x_i} for an individual publisher x_i to join a coalition S as:

$$U_{x_i}(S, \Omega) = G_{x_i}^S - E_{x_i}^S \quad (5.4)$$

where, in the above utility function, G_{x_i} and E_{x_i} are the respective gains and costs function of the publisher x_i forming coalitions. The term Ω is the preference order for a device to enter coalitions and will be discussed later in the chapter.

Similarly, the combined utility $U_S = U_{\{x_i; x_1, x_2, \dots, x_k\}}$ represents the sum of individual publisher's utilities within a coalition and is given by:

$$\begin{aligned} U_S &= \sum_{x_i \in S} U_{x_i}(S, \Omega) = \sum_{x_i \in S} G_{x_i}^S - E_{x_i}^S \\ &= \sum_{x_i \in S} CMI_{x_i}(x_i; x_1, x_2, \dots, x_k, \Delta) - \sum_{x_i \in S} E_{x_i}^S \\ &= G_S(x_i, x_1, x_2, \dots, x_k, \Delta) - \sum_{x_i \in S} E_{x_i}^S \end{aligned}$$

where, the above combined utility takes into account the difference between the corresponding gains $G_{x_i}^S$ and costs $E_{x_i}^S$ of all publishers forming coalitions in the urban environment. In terms of conditional mutual information CMI , the purposed gain function aims

to achieve the highest possible interaction/synergy information G_S inside a coalition with the less energy consumption.

5.2.3 Coalitional Game formation

The coalition formation process is given in [Algorithm 3](#) with the aim of achieving a stable coalition structure. This one aims at forming coalitions based on the preferences of the devices, allowing them to join or leave the coalition at a location l and at a time-slot t , i.e., as a Merge-and-Split algorithm. To leave or join a coalition, each player obeys the principle that it won't make strategy to damage the utility of other players in the original or new coalition.

Definition 1 (Pareto Preference Order)

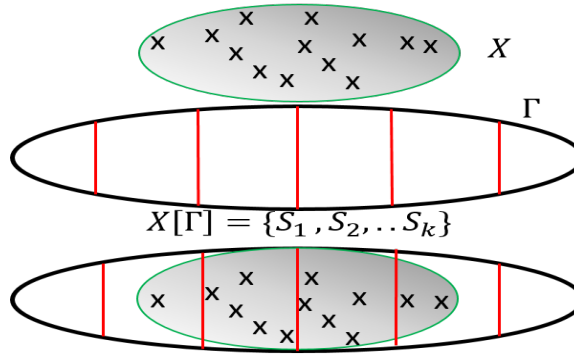
Consider two collections of disjoint coalitions, $\mathbb{S}_a = \{S_{a1}, S_{a2}, \dots, S_{ak}\}$ and $\mathbb{S}_b = \{S_{b1}, S_{b2}, \dots, S_{bk}\}$. The collection \mathbb{S}_a is preferred over \mathbb{S}_b by Pareto order, i.e. $\mathbb{S}_a \triangleright \mathbb{S}_b$ if and only if the way \mathbb{S}_a partition \mathbb{X} is preferred to the way \mathbb{S}_b partition \mathbb{X} . There exists two comparison modes (i) coalition value orders and (ii) individual value orders [62]. Due to the non-transferable nature of the proposed (\mathbb{X}, U) game, we have to use the second comparison relation. To formalize this approach we need the notion of an individual value function $U_{x_i}^{\mathbb{S}}$ as the utility of a player x_i when it belongs to the set of coalitions \mathbb{S} .

$$\mathbb{S}_a \triangleright \mathbb{S}_b \Rightarrow \begin{cases} \forall x_i \in \mathbb{X}, U_{x_i}^{\mathbb{S}_a} \geq U_{x_i}^{\mathbb{S}_b}, \\ \exists x_j \in \mathbb{X}, U_{x_j}^{\mathbb{S}_a} > U_{x_j}^{\mathbb{S}_b} \end{cases}$$

The above Pareto order reflects the behavior of a set of publishers preference to form S_a instead of S_b , if this preference order allows for at least one device to improve its payoff without reducing the utility of any other nearby publisher [62].

5.2.3.1 Collaborative Context at the Edge Algorithm

The Collaborative Context at the Edge Algorithm [Algorithm 3](#) builds coalitions based on the publishers' preferences and allowing only one member to join or leave the coalition

Figure 5.1: Partition function Γ

at a time. Based on the above merge and split definitions, a publisher $x_i \in \mathbb{S}$ already in coalition S_i can join a new coalition S_j in the iteration $v + 1$, if and only if this new coalition S_i was never visited before by the publisher $h_{x_i} \cap S_j = \emptyset$, and the player can promote maximum utility U_{x_i} , while not reducing the utility of any other publisher from previously left and currently joined coalitions respectively S_i, S_j . Similarly, a publisher x_i can split from a coalition S_i to form a singleton S_k if its utility is under a certain *threshold* while the split operation doesn't hurt any of the publisher's utilities from the coalition it left S_j . Hence, a merge or split decision by Pareto order will ensure that all the involved publishers agree on it. The above phases are periodically repeated until convergence, i.e. such that the publishers can autonomously adapt the coalitions structures to environmental changes such as mobility, see Figure 5.2. The granularity of such merge-split operations can be smaller in highly mobile environments to allow a more adequate adaptation of the topology and vice versa in a rather stable topology. Any merge or split operation, results in a new partition of the initial set X as shown in the Figure 5.1, if this new partition is not a D_{hp} Stable partition then the merge and split operations are repeated. In what follows, we present the D_{hp} Stability criteria.

5.2.3.2 Coalition Stability and Convergence

The final partition Γ is \mathbb{D}_{hp} -stable implies that, in this partition, no player or group of devices are interested to perform a merge or a split operation [62] [63].

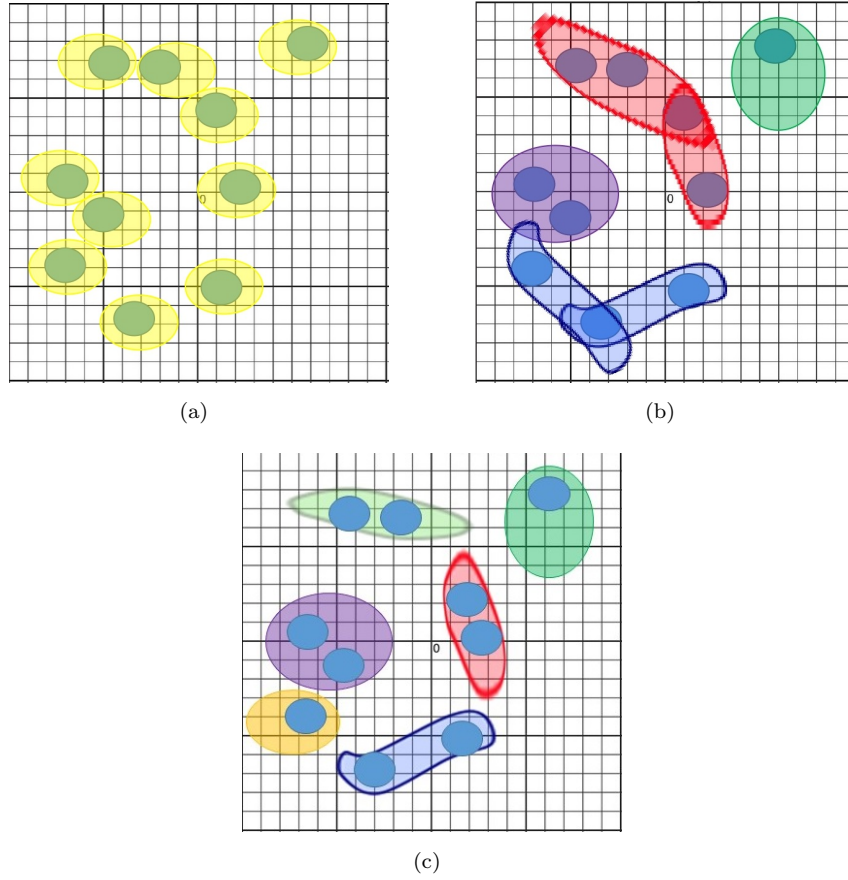


Figure 5.2: Coalition formation through Merge and split operations

Definition 2 (D_{hp} Stability)

A coalition S is \mathbb{D}_{hp} -stable if the following two conditions are satisfied; (i) For each device $x_i \in \mathbb{S}$ and each partition Γ of the coalition, no player or group of players has an interest in performing a merge or a split operation.

Theorem 5.1 ($(\mathbb{D}_{hp}$ Stability)). *Any coalition formed by the proposed in Algorithm 3 is \mathbb{D}_{hp} -stable.*

Proof. (Existence) The proposed dynamic coalitional game converges to the Pareto optimal D_{hp} -stable partition, if such a partition exists [62] [63]. Assume that the final partition Γ is not \mathbb{D}_{hp} -stable, thus a new partition $\Gamma' \triangleright \Gamma$ is preferred based on a Pareto-order criterion. At least one of the publishers can improve its utility while not cutting down the utilities of any other publisher. If the publisher x_k can improve its utility by a merge operation, this implies that the corresponding publisher will leave the coalition S_a to join a coalition S_b . Let's consider that such a node exists and can improve its payoff while not cut down

other nodes utilities. As in immediate consequence of Pareto optimality order used in the [Algorithm 3](#), a merge and/or split operation are operated. \square

Algorithm 3

 Collaborative Context at the Edge Algorithm

INPUT: : The set of discovered publishers ($\mathbb{X} = \{x_1, x_2, \dots, x_n\}$) by Hub at location l and time-slot t .

INITIALIZE: for each discovered publisher $x_i \in \mathbb{X}$ affect a coalition such that the initial partition is: $\Gamma = \{T_1, T_2, \dots, T_n\}$ splits \mathbb{X} into a set of coalitions \mathbb{S} [Figure 5.1](#).

1: **repeat**(2) (3)

2: **Merge** For each player $x_i \in \mathbb{X}$ splits from S_i and joins S_j^* to form a new partition Γ' iff

$$\underset{j}{\operatorname{argmax}} U_{x_i}(S_j, \Omega) = \{U_{x_i}(S_j, \Omega) | \forall S_j \in \mathbb{S} \setminus h_{x_i} : U_{x_i}(S_i, \Omega) < U_{x_i}(S_j, \Omega)\} \quad (5.5a)$$

$$\forall x_j \in S_j, U_{x_j}(S_j, \Omega) \leq U_{x_j}(S_j \cup x_i, \Omega) \Rightarrow U_{S_j} \leq U_{S_j \cup x_i}, \quad (5.5b)$$

$$\text{and} \forall x_k \in S_i, U_{x_k}(S_i, \Omega) \leq U_{x_k}(S_i/x_i, \Omega) \Rightarrow U_{S_i} \leq U_{S_i/x_i}, \quad (5.5c)$$

Then

$$\{S_i, S_j\} \rightarrow \{S_i/x_i, S_j \cup x_i\}, \quad (5.6a)$$

$$h_{x_i} \rightarrow h_{x_i} \cup S_j \quad (5.6b)$$

3: **Split** For each player $x_i \in \mathbb{X}$ select the publisher with the a marginal utility U_{x_i} s.t. $U_{x_i} \leq \sigma$ then x_i splits from S_i to form a new partition Γ' iff

$$\forall x_k \in S_i, U_{x_k}(S_i, \Omega) \leq U_{x_k}(S_i/x_i, \Omega) \Rightarrow U_{S_i} \leq U_{S_i/x_i} \quad (5.7a)$$

Then

$$S_i \rightarrow \{S_i/x_i, S_k | S_k = \{x_i\}\}, \quad (5.8a)$$

$$h_{x_i} \rightarrow h_{x_i} \cup S_k \quad (5.8b)$$

4: **until** Convergence

5: Repeat (1) (2) (3) (4) every θ duration or when any environment change occurs.

From [Theorem 5.1](#), we noticed that the D_{hp} -stable partition is an outcome of the iterative merge-split [Algorithm 3](#) for coalition formation. Therefore, the devices can exploit the merge-split operations to change the coalitions until D_{hp} -stability is achieved. In addition, the merge-split operations in the dynamic coalitional game allows the devices to decide whether they will remain in a coalition to increase their utility in case environmental changes occur. Indeed, the proposed game is repeated periodically in order for devices to make a new merge-split decisions and thereby adapting to the environmental changes.

Theorem 5.2 (\mathbb{D}_c Stability). *A \mathbb{D}_c Stable partition $\Gamma = \{\Gamma_1, ..\Gamma_l\}$ is a unique \triangleright partition, that is for all partitions $\Gamma' \neq \Gamma$ of \mathbb{X} , $\Gamma' \triangleright \Gamma$. In the case, where \triangleright represents the Pareto order, this implies that the \mathbb{D}_c -stable partition Γ is the partition that presents a Pareto optimal utility distribution for all the players. No player or group of players has an incentive to leave this partition using any operation: (i) For each pair of disjoint coalitions S_a and S_b s.t. $\{S_a, S_b\} \subseteq \Gamma$, $(S_a \cup S_b) \triangleright \{S_a, S_b\}$. (ii) For any coalition $R \subseteq \mathbb{X}$ formed of players belonging to different $\Gamma_i \in \Gamma$, $\{R\} [\Gamma] \triangleright \{R\}$. The existence of a \mathbb{D}_c -stable partition is not always guaranteed [62].*

Proof. The first assumption(i) is a direct consequence of the [Theorem 5.2](#). However the second condition (ii) cannot be always satisfied by our algorithm as the super-additivity does not hold by the proposed utility expression [Equation \(5.4\)](#). Indeed, adding more IoT devices into a coalition does not always yield an increase in the utility, it could also increase the cost for information exchange. \square

5.3 Performance Evaluation

We evaluate the performance of the proposed system by generating synthetic data. The target is to find out how well our system can extract and estimate the context of IoT devices while preserving energy. We show that it is feasible to satisfy user requirements, using a less data transmission based on nodes context. We prove that contextualizing IoT devices in an energy efficient manner can reduce energy consumption, furthermore it can reduce the overhead and increase the throughput.

5.3.1 NS3 Simulation-based evaluation

We implement two versions of an energy-efficient content based Publish-Subscribe where one considers the context-awareness based mainly on location and the other is our proposed CEEPS4IoT. Simulations are performed using Network Simulator Version 3 (NS-3)[64] as a scalable simulation platform, considering up to 300 nodes in a neighborhood. To define a neighborhood, we use a set of IEEE 802.11b/g/n enabled nodes in the communication range of each other. In the following simulations, publishers collect data every 2 minutes and the packet size of the payload is set to 123 bytes. Moreover, we consider a simulated human mobility trace with user speeds varying from 0 to 1.5 mps. The mobility scenario is simulated using the "RandomWalk2dMobilityModel" available in NS3, considering an area of $100 \times 100m^2$. We add a radio energy model to each node based on NS3::WifiRadioEnergyModelHelper, considering the following setups: Initial energy as $100J$, the energy consumption for one beacon transmission is $Tr = 0.0174J$ and for switching between active and sleep modes as $0.000426J$. We analyze the scalability of our model with varying the number of nodes as data publishers and subscribers from 50 to 300 in a neighborhood, each simulation is repeated ten times.

5.3.1.1 Implementation

Publish-Subscribe From state of art we conclude that filtering messages based on publisher's context can reduce the energy consumption and latency while increasing the throughput. To compare our work to state of art context-aware Publish-Subscribe systems, we implement a context manager responsible for (i) coalitions formation and update of the context, along with (ii) a sensor selection scheme described below.

Sensor selection scheme The context manager is responsible for tracking the publisher's context. Thus, for each publisher we rank the K top neighbors able to collaborate (i.e. inside the same coalition) based on their utility inside the coalition. Then, the context manager subscribe to this subset of nodes as it plays the role of a context information consumer. When receiving context raw data, the context manager component extract the context information based on any context reasoner (we consider this function out of purpose for this work). Based on our context knowledge, we filter out the publishers in the network by matching their context to the user requirements.

5.3.1.2 Evaluations scenarios

The performance of the proposed model is validated by a set of simulation runs. One of the basic requirements for evaluating the efficiency of a Publish-Subscribe system based our proposed context manager is the user relevant interest satisfied. We consider the following performance metrics:

Energy consumption (Joules): The average Energy consumed by the sensors in the network.

Latency (seconds): The average Latency caused by the message exchanges in the network.

Throughput(%): The successful data delivery rate over communication channels.

We study the effect of varying the network size on the considered location-based Publish/Subscribe and CEEPS4IoT schemes for 50, 100, 200 and 300 nodes. For each configuration, we set the amount of publishers to 50% of the network size and the rest are subscribers (e.g. Network size=100, Publishers=50, Subscribers=50). We consider only two main topics {Temperature, Luminosity} and their related sub-topics{Ambient temperature, temperature under-shade, Ambient luminosity and luminosity under-shade}. The half of the publishers provide services for one of the specified topics and/or sub-topics (e.g. 50 Publishers: 25 publish temperature data, and the rest (25) publish luminosity). Similarly, the subscribers are organized into two equal size set, each interested on a specific topic or any of its related sub-topic {temperature,luminosity}. The publishers depending on their physical context when collecting data, could publish data related to the following sub-topics{Ambient temperature, temperature under-shade, Ambient luminosity, luminosity under-shade}. Similarly the Subscribers, could subscribe for a specif sub-topic. For our proposed schema, we filter publishers based on context matching the user requirements, for example if an application subscribe for temperature under-shade it cannot be provided with ambient temperature data.

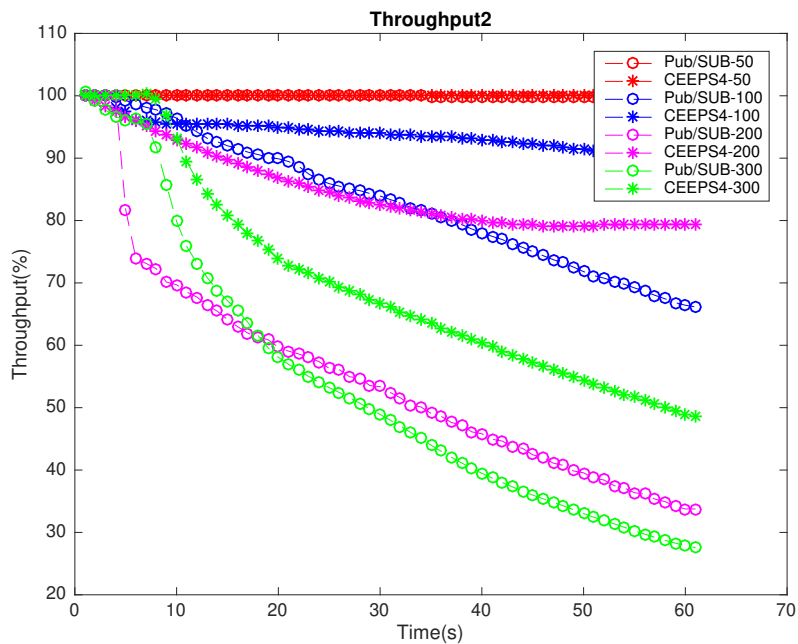


Figure 5.3: Throughput

5.3.2 Evaluation Results

In this section we discuss the evaluation results obtained from our implementation where we study the impact of increasing the number of sensors on the throughput, energy consumption and latency for a network of up to 300 sensors.

5.3.2.1 Throughput

We computed the throughput as a network-oriented metric to evaluate the performance of our proposed coalition game in selecting the best sensors for data collection process. Figure 5.3 shows the throughput (in percentage) achieved by the proposed CEEPS4IoT compared to the benchmark location-based Publish-Subscribe system. The amount of sensors are 50, 100, 200 and 300 in a neighborhood for both approaches. We observe that CEEPS4IoT outperforms the compared approach for all neighborhood size where even in large neighborhood comprising 300 sensors, a throughput of more than 50% is achieved. On the other hand, a throughput of less than 30% is achieved by the existing location-based Publish/Subscribe system.

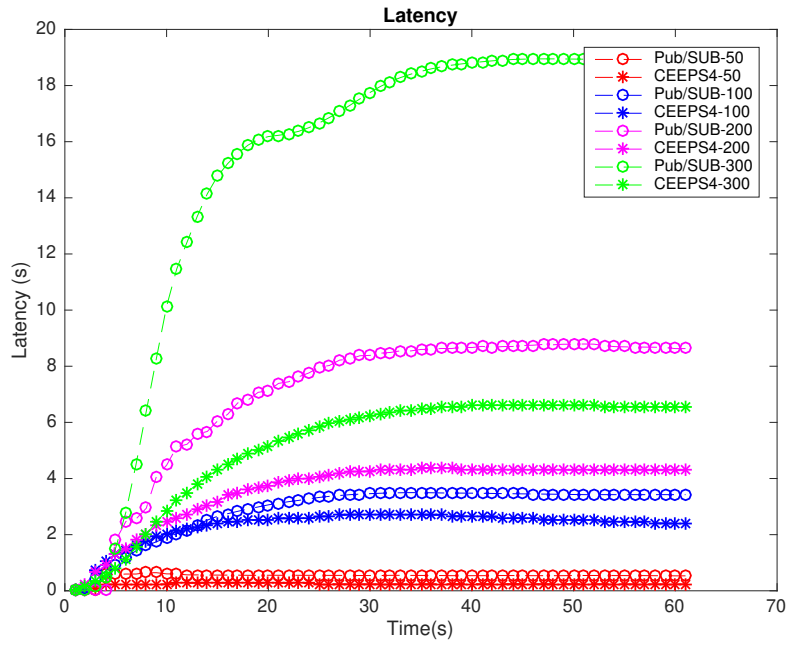


Figure 5.4: Latency

The high throughput of our proposed scheme is due to its consideration of device context while collecting data along incentivizing high quality sensors to cooperate in the data collection process. We notice a decrease in throughput with the increase in neighborhood size for all schemes. This is seen particularly in high neighborhood sizes such as for 300 nodes we notice a larger drop in throughput over time, i.e. for CEEPS4IoT, it dropped from 100% to 50% over time. This is due to the mobility of nodes resulting in difficulty in finding appropriate publishers in the network. Despite the challenging environment, CEEPS4IoT yields a stable throughput over time i.e. around 100% for 50 nodes neighborhood, between 100% and 90% for 100 nodes and between 100% and 85% for 200 nodes. Thus, it achieves higher overall throughput compared to the location-based Publish/Subscribe approach for all neighborhood sizes, thereby, validating its efficiency.

5.3.2.2 Latency

The data availability in a timely manner is important to different IoT applications, therefore, we computed the latency (in seconds) achieved by the proposed coalition game during the simulation. Figure 5.4 depicts the latency incurred by CEEPS4IoT compared to the existing location-based Publish-Subscribe scheme. We observe that the difference between both schemes increases along with the increase in the number of nodes and time duration,

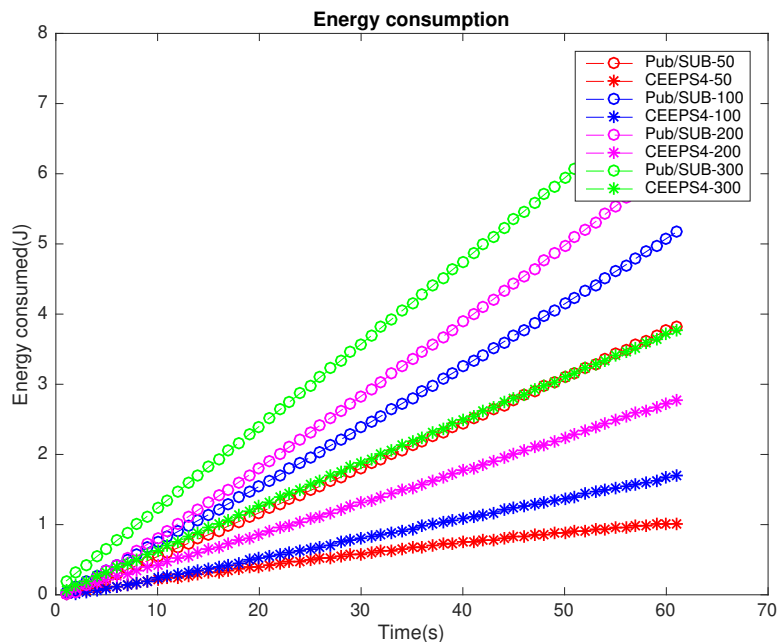


Figure 5.5: Energy

where for 50 nodes the latencies for both schemes are negligible. We can notice that our CEEPS4IoT model can satisfy user requirements (subscriptions) with around two-thirds the time delays generated by location-based Publish-Subscribe for a 100-node network size. For large neighborhood size, CEEPS4IoT reduces the average network latencies by nearly three times compared to location-based Publish-Subscribe system.

This is due to its timely formation of coalitions and contextualization as resulting in lower network overhead and consequently lower delays in responding to the user interests. Contextualization can suppress the redundant and out of a target sensing process. The larger the neighborhood size, the more the latency due to large amount of sensors. This is mainly due to network overhead and nodes mobility. Thus, the overall latency analysis reveals that CEEPS4IoT is an efficient Publish/Subscribe system for IoT with context-awareness.

5.3.2.3 Energy

We computed the energy-consumption as a network-oriented metric to evaluate the performance of our proposed context manager in contextualization of data from resource limited devices. Figure 5.5 shows the energy consumption (in Joules) over time considering

network topologies of 50, 100, 200 and 300 nodes. It is clearly shown that CEEPS4IoT consumes less energy, even when there are large number of sensors present in a neighborhood. With 6x the number of nodes, our model consumes almost the same amount of energy as the location-based Publish-Subscribe framework.

We observe an increase on the energy consumption along with the increase of the network size $|\mathcal{S}|$ for both models, however the energy consumption of CEEPS4IoT can save from almost 50% and 75% of the total energy for the same task for 300-nodes and 50-nodes, respectively.

We notice that the energy consumption increases over time irrespectively of the number of sensors. This is due to the increase in the amount of messages exchanged between nodes. Overall, the energy consumption analysis results suggest that our context-aware Publish-Subscribe using our collaborative approach is indeed energy efficient.

5.4 Conclusions

Large number of sensors constantly updating data in an urban environment can be an energy wasting operation resulting in a large amount of redundant and unwanted data. Data Provider instruct data collection from different sensors/devices, unaware of their source's context. We propose in this work to rethink data collection and to limit the data collection to the tiny amount of sensors that meet the application needs. This chapter proposes an energy efficient and context-aware publish subscribe system *CEEPS4IoT* that orchestrate the data collection based on the physical sensors context. We allow a data provider to keep track of publisher's context based on a collaborative strategy between co-located devices, in an efficient manner. However, devices are reluctant to share information due to energy cost involved by this operation. To cater this, we based our solution on a dynamic coalitional game that incentivise devices to receive as a gain a reward which is proportionally to the the context information shared with neighbors along the compensation of their incurred cost.

We evaluate the proposed *CEEPS4IoT* using NS3-simulations, on different network sizes, i.e. 50, 100, 200 and 300 mobile nodes. For each topology we consider 50% of the nodes as

publisher and the rest as subscribers, publishing and subscribing for two topics (temperature, luminosity). We consider the existence of only one broker for each tested topology, supporting a Publisher message transmission frequency of 1 packet/sec. Extensive simulations and results show that our scheme is a low-latency and energy efficient. More precisely, our proposed scheme can reduce energy consumption by 50% to up-to 75%, and the network latency by 30% to up-to 70% in comparison to the location-based Publish-Subscribe scheme. In addition, *CEEPS4IoT* can achieve almost 50% of throughput compared to only 30% for the location-based scheme.

Chapter 6

Conclusion and Future works

Contents

6.1 Contributions	95
6.1.1 Context Reasoner	96
6.1.2 WELCOME	96
6.1.3 CEEPS4IoT	97
6.2 Future Works	98

6.1 Contributions

The main objective of this thesis is to present an efficient paradigm of “T2T context awareness at the edge” which allows IoT device/services to become aware of their physical context. Through all the contributions, the present thesis reached the following objectives:

- IoT devices situational context analysis.
- Optimized resource consumption for a low-latency neighbor discovery.
- IoT device’s context-driven Publish/Subscribe.

In the following we summarize our three main contributions, encountered challenges and results.

6.1.1 Context Reasoner

We firstly introduce the core component of “T2T context aware computing” which is a context reasoner based on a collaborative intelligence between nearby devices in the same vicinity. It processes ongoing sensory data from neighbor devices to extract knowledge about the sequence of contexts IoT device(s) passed through during its/their lifetime. This reasoner act like human brain by learning the spatio-temporal IoT devices relations, and is robust enough to irrelevant nearby observations thanks to its penalization functions.

In IoT, observations of the environment (e.g. temperature, luminosity...) exhibit a spatio-temporal correlation as such physical conditions rarely occur independently. We proposed a Spatio-temporal CRF, that models such spatio-temporal relations between nearby sensor readings and also between previous and actual device states. As existing spatio-temporal CRF model doesn't meet IoT needs, we proposed a spatio-temporal CRF model that consider separately the spatial and temporal potentials functions, as they could have different impacts on context estimation depending on the application domain.

We evaluated our model with different scenarios on real and synthetic data-sets, to study the influence of the increase of neighborhood size, neighbor distances and lagtime. Our proposed model was evaluated on up to 200 test-bed context sequences each composed by 10 contexts to estimate, and reached around 98.5% of accuracy when there is enough close neighbors and recent sensory data. In case our model is confronted to more than 80% biased data back dating to one hour ago, it is still able to accurately estimate the context sequences with almost 70% of accuracy. We can summarize that our CRF-based-reasoner can perform very well in case there exists neighbors that are fairly close, reporting recent observations. This one is also quite robust regarding biased and noisy data.

6.1.2 WELCOME

We proposed “WELCOME”, a novel neighbor discovery scheme allowing a single node as a delegate, instead of all nodes in a proximity, to send beacons and listen to neighbors thereby reducing the overall amount of transmitted messages. This cater the issue of collisions between messages sent by concurrently transmitting nodes. We further allow nodes to

rotate the role of becoming the delegate where each node computes its eligibility to become the delegate node based on its residual energy and association to the neighborhood. Moreover, WELCOME enables nodes to self-organize where eligible nodes contend to become delegate in the absence of an existing delegate node.

We defined three metrics to study the relation between *i*) the energy consumption and the fraction of neighbors discovered, and *ii*) the latency of discovery and the fraction of neighbors discovered. The purpose is to analyze WELCOME with respect to the energy efficiency and latency trade-off, i.e. reducing the energy consumption by employing low duty cycles can lead to longer latency in discovering neighbors.

WELCOME is evaluated theoretically as well as using simulations, analyzing its scalability to discover up to 100 neighbors in each other communication range for nodes operating on low duty cycles of 5%. Results show that WELCOME is a low latency and energy efficient neighbor discovery scheme allowing nodes to successfully discover 100% of their neighbors unlike existing schemes which resulted in discovering a maximum of 90% nodes with substantially high latency. The contributions of this chapter are summarized as follows:

6.1.3 CEEPS4IoT

At least, we presented *Context-aware Energy Efficient Publish-Subscribe for Internet of Things (CEEPS4IoT)* scheme. It is a novel Publish-Subscribe platform that enables service providers to be aware of the context of objects in an energy efficient manner while collecting data in urban environment. We proposed filtering and aggregation of sensor data on mobile devices prior to its transmission into the cloud based on data requirements, as a solution to limit data transmission to only information of interest. IoT devices in the same location provide information about different aspects of the environment, thus, we can leverage their coexistence to infer their respective contexts. Precisely, we measured the relation of a device with the readings from nearby sensors with the goal to capture the amount of information they can provide regarding its environment. The idea is to find out whether a sensor should be selected to assist an IoT application by taking into account the impact of the environment on the collected data.

First, we modeled the interest for objects, simultaneously present in the same physical location, to share readings regarding each others context by the interaction information. Then, we proposed a dynamic coalitional game for resource limited [48] objects as publishers to provide necessary readings for context knowledge extraction under the energy consumption constraint. Objects can merge into or split from spatio-temporal coalitions based on the payoff each coalition provides where it maintains a preference ordering among coalitions to join, in order to facilitate an energy efficient and high quality data collection. It is however challenging to maximize the information about each object's context while avoiding extra energy consumption. We propose as a reward for each object to enter a coalition, a utility function proportional to the the amount of data nearby objects shares about its context and inversely proportional to the the energy cost of the amount of data it has to exchange. Further, we provide the stability and convergence conditions for our dynamic coalitional game along relevant proofs for its existence and uniqueness.

CEEPS4IoT is evaluated using extensive simulations and results reveal that it is an energy efficient Publish-Subscribe system for IoT. Results from evaluating CEEPS4IoT in networks of upto 300 nodes suggest it as a scalable and energy efficient location-based Publish-Subscribe system with context awareness. Our proposed scheme saves energy consumption by 50% to up-to 75%, moreover it reduces the network latency by 30% to up-to 70% compared to the selected Pub/Sub model. In addition, we found that among all the compared schemes CEEPS4IoT reached almost 20% additional throughput.

6.2 Future Works

In addition to the contributions presented in this manuscript, the work done during this thesis also gave essence to some direct future research perspectives. The first one includes the extension of the CRF-based model to avoid any expert knowledge intervention. The objective here would be to provide a framework which could avoid expert tuning by learning the interaction and association functions. The aim will thus be to extend our model by the addition of a deep component upstream to the proposed CRF component. This step will allow to efficiently and accurately estimate the context of IoT devices without any prior knowledge. Moreover, the plan here would be to extend the context features with more

than the physical environment conditions, to include other features related to the network, the operating system, the devices lifetime, and so on allowing for an even better selection.

The second perspective concerns WELCOME, this scheme, ensures for an IoT device located in an opportunistic and self-organized network to discover close neighbors if they exist while better preserving energy. Possible extensions are to study the relation between the neighborhood size and the amount of time a delegate listens for messages from neighbors. Our future work here is to allow delegate nodes to adapt their listen period by learning the neighborhood size based on the amount of received Hello messages. This will result in further reduction in the delay of discovering neighbors as well as the amount of idle listening period for a delegate node. The best trade-off delay-energy can thus be achieved.

Concerning the last contribution, CEEPS4IoT, we plan to extend our study considering the case of a large scale network and more precisely highly dynamic networks. For that need we can evaluate our scheme based on realistic mobility traces. Managing IoT devices mobility and context sharing, as a solution for highly mobile devices. Moreover, we designed CEEPS4IoT for a semi-centralized architecture with a mobile broker to filter-out data near to data sources. Thus, we propose to consider a fully distributed networks, using a Coalition Graph Game. In addition, the utility for each player can take into account more than the synergy/mutual information. For instance, we can consider features as the stability of links and the quality of service in order to better ensure an accurate context estimation and at the same time fast convergence to stable partitions.

The proposed concept of “T2T context-awareness at the edge computing” is wider than just the propositions presented earlier, even having these are major ones. Indeed, making Things interactions and communications context-aware, offers several advantages leading to many others mid-term research perspectives. One of these perspectives would be to propose to enhance the Quality of Context with a semantic QoC parameter: We can assign to each data a semantic QoC parameter, that provide a semantic description to the numerical data. How to make this assignment is an open issue.

List of Publications

International Conferences

- **HARMASSI, Mariem, FAUCHER, Cyril, et GHAMRI-DOUDANE, Yacine.** *Smart things: Conditional random field based solution for context awareness at the IoT edge.* In : 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2018. p. 1-7.
- **HARMASSI, Mariem, KHAN, Junaid Ahmed, GHAMRI-DOUDANE, Yacine, et al.** *Welcome: Low Latency and Energy Efficient Neighbor Discovery for Mobile and IoT Devices.* In : 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE, 2018. p. 1-9.
- **HARMASSI, Mariem, FAUCHER, Cyril, et GHAMRI DOUDANE, Yacine.** (6.0) *CEEPS4IoT: Context-aware Energy Efficient Publish/Subscribe for Internet of Things.* In : Proceedings of the 14th ACM International Symposium on QoS and Security for Wireless and Mobile Networks. ACM, 2018. p. 85-91.

Awards

- Runner-up to IEEE BEST PAPER AWARD, CCNC, 2018 <https://ccnc2018.ieee-ccnc.org/program/awards>

References

- [1] Mariem Harmassi, Cyril Faucher, and Yacine Ghamri-Doudane. Smart things: Conditional random field based solution for context awareness at the iot edge. In *Consumer Communications & Networking Conference (CCNC), 2018 15th IEEE Annual*, pages 1–7. IEEE, 2018.
- [2] Mariem Harmassi, Junaid Ahmed Khan, Yacine Ghamri-Doudane, and Cyril Faucher. Welcome: Low latency and energy efficient neighbor discovery for mobile and iot devices. In *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–9. IEEE, 2018.
- [3] Mariem Harmassi, Cyril Faucher, and Yacine Ghamri Doudane. Ceeps4iot: Context-aware energy efficient publish/subscribe for internet of things. In *Proceedings of the 14th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, pages 85–91. ACM, 2018.
- [4] Christine Julien, Chenguang Liu, Amy L Murphy, and Gian Pietro Picco. Blend: practical continuous neighbor discovery for bluetooth low energy. In *IPSN*, pages 105–116, 2017.
- [5] P Kindt, D Yunge, G Reinerth, and S Chakraborty. Griassdi: Mutually assisted slotless neighbor discovery protocols. In *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2017.
- [6] Wei Sun, Zheng Yang, Keyu Wang, and Yunhao Liu. Hello: A generic flexible protocol for neighbor discovery. In *INFOCOM, 2014 Proceedings IEEE*, pages 540–548. IEEE, 2014.
- [7] Ying Qiu, Shining Li, Xiangsen Xu, and Zhigang Li. Talk more listen less: Energy-efficient neighbor discovery in wireless sensor networks. In *Computer Communications*,

- IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016.
- [8] Mehedi Bakht, Matt Trower, and Robin Hilary Kravets. Searchlight: Won't you be my neighbor? In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 185–196. ACM, 2012.
- [9] Keyu Wang, Xufei Mao, and Yunhao Liu. Blinddate: A neighbor discovery protocol. *IEEE Transactions on Parallel and Distributed Systems*, 26(4):949–959, 2015.
- [10] Michael J McGlynn and Steven A Borbash. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 137–145. ACM, 2001.
- [11] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, WMCSA '94, pages 85–90, Washington, DC, USA, 1994. IEEE Computer Society.
- [12] Peter J Brown, John D Bovey, and Xian Chen. Context-aware applications: from the laboratory to the marketplace. *IEEE personal communications*, 4(5):58–64, 1997.
- [13] N Ryan, J Pascoe, and D Morse. Enhanced reality fieldwork: the context aware archaeological assistant. In *Archaeology in the age of the internet: CAA 97: computer applications and quantitative methods in archaeology: proceedings of the 25th anniversary conference*, page 75, 1997.
- [14] Anind K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5(1):4–7, January 2001.
- [15] Tarak Chaari, Frédérique Laforest, and André Flory. Adaptation des applications au contexte en utilisant les services web. In *Proceedings of the 2nd French-speaking conference on Mobility and ubiquity computing*, pages 111–118. ACM, 2005.
- [16] Jing He, Yanchun Zhang, Guangyan Huang, and Jinli Cao. A smart web service based on the context of things. *ACM Trans. Internet Technol.*, 11(3):13:1–13:23, 2012.
- [17] Qiang Wei and Zhi Jin. Service discovery for internet of things: a context-awareness perspective. In *Proceedings of the Fourth Asia-Pacific Symposium on Internetware*, page 25. ACM, 2012.

-
- [18] Simon Mayer, Dominique Guinard, and Vlad Trifa. Searching in a web-based infrastructure for smart things. In *Internet of Things (IOT), 2012 3rd International Conference on the*, pages 119–126. IEEE, 2012.
- [19] Giuseppe Fenza, Domenico Furno, and Vincenzo Loia. Hybrid approach for context-aware service discovery in healthcare domain. *Journal of Computer and System Sciences*, 78(4):1232–1247, 2012.
- [20] Naoya Namatame, Yong Ding, Till Riedel, Hideyuki Tokuda, Takashi Miyaki, and Michael Beigl. A distributed resource management architecture for interconnecting web-of-things using ubox. In *Proceedings of the Second International Workshop on Web of Things*, page 4. ACM, 2011.
- [21] G r me Bovet and Jean Hennebert. Distributed semantic discovery for web-of-things enabled smart buildings. In *New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on*, pages 1–5. IEEE, 2014.
- [22] Charith Perera, Arkady Zaslavsky, Peter Christen, Michael Compton, and Dimitrios Georgakopoulos. Context-aware sensor search, selection and ranking model for internet of things middleware. In *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, volume 1, pages 314–322. IEEE, 2013.
- [23] Yao-Chung Hsu, Chi-Han Lin, and Wen-Tsuen Chen. Design of a sensing service architecture for internet of things with semantic sensor selection. In *Ubiquitous Intelligence and Computing, 2014 IEEE 11th Intl Conf on and IEEE 11th Intl Conf on and Autonomic and Trusted Computing, and IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UTC-ATC-ScalCom)*, pages 290–298. IEEE, 2014.
- [24] Talal Ashraf Butt, Iain Phillips, Lin Guan, and George Oikonomou. Trendy: An adaptive and context-aware service discovery protocol for 6lowpans. In *Proceedings of the third international workshop on the web of things*, page 2. ACM, 2012.
- [25] Farshid Hassani Bijarbooneh, Wei Du, Edith C-H Ngai, Xiaoming Fu, and Jiangchuan Liu. Cloud-assisted data fusion and sensor selection for internet of things. *IEEE Internet of Things Journal*, 3(3):257–268, 2016.

-
- [26] Zhichuan Huang, Tiantian Xie, Ting Zhu, Jianwu Wang, and Qingquan Zhang. Application-driven sensing data reconstruction and selection based on correlation mining and dynamic feedback. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 1322–1327. IEEE, 2016.
- [27] João B Borges Neto, Thiago H Silva, Renato Martins Assunção, Raquel AF Mini, and Antonio AF Loureiro. Sensing in the collaborative internet of things. *Sensors*, 15(3):6607–6632, 2015.
- [28] Ruben Verborgh, Thomas Steiner, Davy Van Deursen, Rik Van de Walle, and Joaquim Gabarró Vallés. Efficient runtime service discovery and consumption with hyperlinked restdesc. In *2011 7th International Conference on Next Generation Web Services Practices*, pages 373–379. IEEE, 2011.
- [29] Ali Yachir, Yacine Amirat, Abdelghani Chibani, and Nadjib Badache. Event-aware framework for dynamic services discovery and selection in the context of ambient intelligence and internet of things. *IEEE Transactions on Automation Science and Engineering*, 13(1):85–102, 2016.
- [30] Lin Chen and Kaigui Bian. Neighbor discovery in mobile sensing applications: A comprehensive survey. *Ad Hoc Networks*, 48:38–52, 2016.
- [31] Ricardo C Carrano, Diego Passos, Luiz CS Magalhães, and Célio VN Albuquerque. A comprehensive analysis on the use of schedule-based asynchronous duty cycling in wireless sensor networks. *Ad Hoc Networks*, 16:142–164, 2014.
- [32] Desheng Zhang, Tian He, Yunhuai Liu, Yu Gu, Fan Ye, Raghu K Ganti, and Hui Lei. Acc: generic on-demand accelerations for neighbor discovery in mobile applications. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 169–182. ACM, 2012.
- [33] Desheng Zhang, Tian He, Fan Ye, Raghu Ganti, and Hui Lei. Neighbor discovery and rendezvous maintenance with extended quorum systems for mobile applications. *IEEE Transactions on Mobile Computing*, 2016.
- [34] Liangyin Chen, Yuanchao Shu, Yu Gu, Shuo Guo, Tian He, Fan Zhang, and Jiming Chen. Group-based neighbor discovery in low-duty-cycle mobile sensor networks. *IEEE Transactions on Mobile Computing*, 15(8):1996–2009, 2016.

-
- [35] Shouwen Lai, Binoy Ravindran, and Hyeonjoong Cho. Heterogenous quorum-based wake-up scheduling in wireless sensor networks. *IEEE Transactions on Computers*, 59(11):1562–1575, 2010.
- [36] Prabal Dutta and David Culler. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 71–84. ACM, 2008.
- [37] Arvind Kandhalu, Karthik Lakshmanan, and Ragunathan Raj Rajkumar. U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol. In *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*, pages 350–361. ACM, 2010.
- [38] Lin Chen, Kaigui Bian, and Meng Zheng. Never live without neighbors: From single-to multi-channel neighbor discovery for mobile sensing applications. *IEEE/ACM Transactions on Networking*, 24(5):3148–3161, 2016.
- [39] Tong Meng, Fan Wu, Aijing Li, Guihai Chen, and Nitin H Vaidya. On robust neighbor discovery in mobile wireless networks. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, page 38. ACM, 2015.
- [40] Tingpei Huang, Haiming Chen, Yuqing Zhang, and Li Cui. Easind: Effective neighbor discovery algorithms for asynchronous and asymmetric-duty-cycle multi-channel mobile wsns. *Wireless Personal Communications*, 84(4):3031–3055, 2015.
- [41] Tong Meng, Fan Wu, and Guihai Chen. On designing neighbor discovery protocols: A code-based approach. In *INFOCOM, 2014 Proceedings IEEE*, pages 1689–1697. IEEE, 2014.
- [42] Junaid Ahmed Khan, Romain Pujol, Razvan Stanica, and Fabrice Valois. On the energy efficiency and performance of neighbor discovery schemes for low duty cycle iot devices. In *ACM 14th PE-WASUN’17*, 2017.
- [43] Patrick Th Eugster, Pascal A Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM computing surveys (CSUR)*, 35(2):114–131, 2003.

-
- [44] Gianpaolo Cugola, Alessandro Margara, and Matteo Migliavacca. Context-aware publish-subscribe: Model, implementation, and evaluation. In *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, pages 875–881. IEEE, 2009.
- [45] Aleksandar Antonić, Martina Marjanović, Krešimir Pripuzić, and Ivana Podnar Žarko. A mobile crowd sensing ecosystem enabled by cupus: Cloud-based publish/subscribe middleware for the internet of things. *Future Generation Computer Systems*, 56:607–622, 2016.
- [46] Heng Wang, Daijin Xiong, Ping Wang, and Yuqiang Liu. A lightweight xmpp publish/subscribe scheme for resource-constrained iot devices. *IEEE Access*, 5:16393–16405, 2017.
- [47] Akram Hakiri, Pascal Berthou, Aniruddha Gokhale, and Slim Abdellatif. Publish/subscribe-enabled software defined networking for efficient and scalable iot communications. *IEEE communications magazine*, 53(9):48–54, 2015.
- [48] Z Shelby, K Hartke, C Bormann, and B Frank. Constrained application protocol (coap), draft-ietf-core-coap-13. *Orlando: The Internet Engineering Task Force-IETF*, 2012.
- [49] D Locke. Mq telemetry transport (mqtt) v3. 1 protocol specification. ibm developerworks technical library (2010), 2010.
- [50] Everton de Matos, Leonardo Albernaz Amaral, Ramão Tiago Tiburski, Matheus Crespi Schenfeld, Dario FG de Azevedo, and Fabiano Hessel. A sensing-as-a-service context-aware system for internet of things environments. In *Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual*, pages 724–727. IEEE, 2017.
- [51] Alfonso Garcia-de Prado, Guadalupe Ortiz, and Juan Boubeta-Puig. Collect: Collaborative context-aware service oriented architecture for intelligent decision-making in the internet of things. *Expert Systems with Applications*, 85:231–248, 2017.
- [52] Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, pages 93–128, 2006.

-
- [53] Christopher N Ververidis and George C Polyzos. Service discovery for mobile ad hoc networks: a survey of issues and techniques. *IEEE Communications Surveys & Tutorials*, 10(3), 2008.
- [54] Charith Perera, Arkady Zaslavsky, Chi Harold Liu, Michael Compton, Peter Christen, and Dimitrios Georgakopoulos. Sensor search techniques for sensing as a service architecture for the internet of things. *IEEE Sensors Journal*, 14(2):406–420, 2014.
- [55] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [56] Thorsten Hoberg, Franz Rottensteiner, Raul Queiroz Feitosa, and Christian Heipke. Conditional random fields for multitemporal and multiscale classification of optical satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 53(2):659–673, 2015.
- [57] BK Kenduywoa, D Bargiel, and U Soergel. Spatial-temporal conditional random fields crop classification from terrasars-x images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3):79, 2015.
- [58] Stephen Wright and Jorge Nocedal. Numerical optimization. *Springer Science*, 35:67–68, 1999.
- [59] Godfrey Harold Hardy, Edward Maitland Wright, et al. *An introduction to the theory of numbers*. Oxford university press, 1979.
- [60] Mariem Harmassi, Cyril Faucher, and Yacine Ghamri-Doudane. Smart things: Conditional random field based solution for context awareness at the iot edge. In *Consumer Communications & Networking Conference (CCNC), 2018 15th IEEE Annual*, pages 1–7. IEEE, 2018.
- [61] Philipp H Kindt, Marco Saur, Michael Balszun, and Samarjit Chakraborty. Neighbor discovery latency in ble-like protocols. *IEEE Transactions on Mobile Computing*, 17(3):617–631, 2017.
- [62] Krzysztof R Apt and Andreas Witzel. A generic approach to coalition formation. *International Game Theory Review*, 11(03):347–367, 2009.

-
- [63] Krzysztof R Apt and Tadeusz Radzik. Stable partitions in coalitional games. *arXiv preprint cs/0605132*, 2006.
- [64] NS-3. Network simulator. [https://https://www.nsnam.org/](https://www.nsnam.org/), 2010.

