



**HAL**  
open science

# Communicating materials : modeling and management of energy consumption

Hang Wan

► **To cite this version:**

Hang Wan. Communicating materials : modeling and management of energy consumption. Ubiquitous Computing. Université de Lorraine, 2021. English. NNT : 2021LORR0131 . tel-03395856

**HAL Id: tel-03395856**

**<https://hal.univ-lorraine.fr/tel-03395856>**

Submitted on 22 Oct 2021

**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.



## AVERTISSEMENT

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

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

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

Contact : [ddoc-theses-contact@univ-lorraine.fr](mailto:ddoc-theses-contact@univ-lorraine.fr)

## LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

[http://www.cfcopies.com/V2/leg/leg\\_droi.php](http://www.cfcopies.com/V2/leg/leg_droi.php)

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

---

# Matériaux communicants : modélisation et gestion de la consommation d'énergie

## THÈSE

Présentée et soutenue publiquement le 13 Septembre 2021

Pour l'obtention du

**Doctorat de l'Université de Lorraine**

**(en Automatique, Traitement du Signal et des Images, Génie Informatique)**

Par Hang WAN

### Composition du jury

#### *Rapporteurs*

- Pr. Jean-Marc THIRIET, Université de Grenoble Alpes
- Pr. Yves SALLEZ, Université Polytechnique Hauts-de-France

#### *Examineurs*

- Pr. Nathalie JULIEN, Université de Bretagne Sud
- Pr. Hind BRIL EL HAOUZI, Université de Lorraine

#### *Encadrants*

- Pr. William DERIGENT, Université de Lorraine (Directeur de thèse)
- Dr. Michaël DAVID, Université de Lorraine (Co-encadrant)

---

Centre National de la Recherche Scientifique (CNRS)

Centre de Recherche en Automatique de Nancy (CRAN) – UMR7039





# Communicating materials: modeling and management of energy consumption

## DOCTORAL THESIS

Submitted in total fulfilment of the requirements for the degree of

**Doctor of University of Lorraine**

**(in Automatic, Signal and Image Processing, Computer Engineering)**

By Hang WAN

September, 2021

### **Members of the jury**

#### *Reviewers*

- Pr. Jean-Marc THIRIET, University of Grenoble Alpes
- Pr. Yves SALLEZ, Polytechnic University of Hauts-de-France

#### *Examiners*

- Pr. Nathalie JULIEN, University of South Brittany
- Pr. Hind BRIL EL HAOUZI, University of Lorraine

#### *Supervisors*

- Pr. William DERIGENT, University of Lorraine (Director of thesis)
- Dr. Michaël DAVID, University of Lorraine (Co-supervisor)

---

National Center of Scientific Research (CNRS)

Center of Research in Automatic in Nancy (CRAN) – UMR7039





# Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Prof. William DERIGENT for his support and encouragement throughout my PHD study. His passion and rigor for science led me to the new world. Since the first day at CRAN, he has provided me with enthusiastic help and invaluable suggestions. I have learned many things from him, especially, the logical way of thinking which will accompany all my following research life.

I also express my warm thanks to my co-supervisor Dr. Michaël DAVID. He has shared a lot of novel ideals and provided professional insights. Besides, he has spent much precious time for reading and improving the quality of our contributions. My work could not be achieved without his patient and careful guide.

Additionally, I would also like to thank all members at CRAN, especially my dear colleagues of ISET team for sharing their experiences on studies and life. They accompanied me to explore different cultures and enriched my life. I am also grateful to administrative staffs at CRAN, especially, the secretaries for their instant support.

Moreover, I would like to extend my thanks to the jury members for spending their precious time to read and review this thesis. I am greatly thankful for the funding of the French National Research Agency and the support of all partners of MCBIM project.

Finally, my thanks would go to my beloved parents for their loving considerations and great confidence. They have always been with me, shared and supported me during the difficult course of the thesis, especially, the period of COVID-19 pandemic.



# Abstract

With the popularization of the Internet of Things, more and more intelligent products appear in our daily life. The McBIM project (Materials communicating with the Building Information Modelling) aims to design a new type of “communicating concrete”: concrete elements equipped with embedded wireless sensor network (WSN). This communicating concrete can measure the physical environment, store the information and exchange data with BIM platforms. One objective is to demonstrate the usefulness of the concept over the whole building lifecycle and the energy management is one of the facing challenges. Meanwhile, the product, user information as well as the large amount of monitoring data bring challenges regarding data management. To solve both issues, this work proposes to couple the physical communicating concrete to an external, digital and intelligent system in order to extend network lifetime.

In this thesis, these physical and digital parts are both studied. In the physical part, energy saving methods are reviewed, to understand how to extend the network lifetime. Network routing, in-network data processing and time synchronization are considered to build an energy model for periodic data collection. This energy model is validated with an Arduino-XBee based platform. The experiment shows that the analytic energy estimation model provides an approximate but accurate remaining energy level. It can be used online by nodes or remotely by user to the remaining energy. Energy-related data and collection modes can be managed by the digital part, based on a multi-agent system (MAS).

The MAS can not only store monitoring data of communicating elements but also provide energy efficient data collection or organization strategies. Composition of communicating elements may happen in construction or exploitation lifecycle’s phase to reduce energy consumption. A recursive multi-agent architecture is proposed to meet high flexibility requirements. This multi-agent system is implemented on a 3D simulation platform to visualize the communicating concrete and verify the usefulness of our analytic energy model. The recursive multi-agent architecture provides energy efficiency solutions to extend network lifetime and respond to environmental changes.

Keywords: Communicating Materials, Wireless Sensor Network, In-Network Processing, Energy consumption model, Multi-Agent System



# Résumé

Avec la popularisation de l'Internet des Objets, de plus en plus de produits intelligents apparaissent dans notre vie. Le projet McBIM (Matière communicante au service du BIM-Building Information Modelling) vise à concevoir un nouveau type de « béton communicant » qui est le béton équipé d'un réseau de capteurs sans fil embarqué (RSCF). Ce béton communicant est capable de mesurer des phénomènes physiques, de stocker des informations et d'échanger des données avec des plateformes BIM. Pour démontrer l'utilité du concept sur l'ensemble du cycle de vie, l'exigence de service à long terme pose des défis relatifs à la durée de vie du réseau intégré dans la matière. Dans le même temps, la grande quantité de données de surveillance posent des défis relatifs à la gestion d'informations. Dans ces travaux, nous proposons de coupler le béton communicant physique à sa représentation digitale. Ce système externe et intelligent doit pouvoir permettre à terme de prolonger la durée de vie du réseau.

La partie physique étudie des méthodes d'économie d'énergie applicables à la collecte de données dans les réseaux de capteurs, afin de comprendre comment prolonger la durée de vie de réseau et construire un modèle de consommation d'énergie du réseau. Le routage du réseau, le traitement des données dans le réseau (In-Network Processing) et la synchronisation des noeuds sont pris en compte pour construire un modèle de consommation énergétique du réseau embarqué, modèle validé avec une plate-forme Arduino-XBee. L'expérience montre que le modèle d'estimation de l'énergie fournit une approximation précise de l'énergie restante, pouvant être utilisée en ligne par les nœuds ou à distance par un utilisateur. La gestion de l'énergie du réseau et des modes de collecte peut-être réalisée par un système multi-agents dédié.

La partie digitale est construite avec un système multi-agents, pouvant non seulement stocker les données de surveillance, mais également fournir des stratégies de collecte de données. La composition des éléments communicants peut intervenir en phase de construction ou d'exploitation. Une architecture multi-agents récursive est proposée pour répondre à cette exigence de flexibilité. Ce système multi-agents est implémenté sur une plateforme de simulation 3D pour visualiser le béton communicant et vérifier la validité de notre modèle énergétique analytique. L'architecture multi-agents récursive fournit des solutions efficaces pour estimer l'énergie résiduelle dans la matière, afin de prolonger la durée de vie du réseau et répondre aux changements environnementaux.

Mots-clés : Matériaux communicants, Réseau de Capteurs Sans Fil, Traitement en réseau, Modèle de consommation d'énergie, Système multi-agents



# LIST OF ACRONYMS

ACK	Acknowledgment
ADACOR	Adaptive holonic control architecture
AMRP	Average minimum reachability power
ANR	French national research agency
BCBRP	Balanced chain-based routing protocol
BOL	Beginning of Life
BS	Base station
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CBD	Composed-Based Development.
CM	Communicating materials
CN	Communicating node
COSA	Coalition oriented sensing algorithm
CRAN	Centre de Recherche en Automatique de Nancy
CRBCC	Chain-cluster based mixed routing
CSESM	Communication subsystem energy consumption model
DMTS	Delay measurement time synchronization
DWEHC	Distributed Weight-based Energy-efficient Hierarchical Clustering protocol
EADAT	Energy-aware distributed aggregation tree
EOL	End of Life
EPC	Electronic product code
EPCIS	EPC information service
HCBA	Holonic component-based architecture
HEED	Hybrid energy-efficient distributed clustering
HMS	Holonic manufacturing system
IDE	Integrated development environment
IFS	Inter-frame space
IoT	Internet of things
IP	Intelligent product
IMS	Intelligent manufacturing systems
LAAS	Laboratory for Analysis and Architecture of Systems
LIB	Computer science laboratory of Burgundy

MAC	Media access control
MAS	Multi-Agent System
MAS-R	Recursive multi-agent system
McBIM	Material communicating with the BIM - Building Information Modelling
MOL	Middle of Life
MST	Minimum spanning tree
NC	Nœud de communication
NEM	Node energy model
NTP	Network time protocol
NS	Nœud sensitif
ONS	Object naming service
PEDAP	Power Efficient Data gathering and Aggregation Protocol
PEGASIS	Power Efficient Gathering in Sensor Information Systems
PEM	Processors energy model
PI	Physical internet
PLM	Product lifecycle management
PMI	PROMISE messaging interface
PROSA	Product-Resource-Order-Staff
PROSIS	Product, Resource, Order, Simulation for Isoarchy Structure
RBS	Reference broadcast synchronization
RCSF	Réseau de capteurs sans fil
RFID	Radio frequency identification
SEM	Sensor energy model
SMA	Système multi-agents
SN	Sensing node
TDMA	Time division multiple access
TEM	Transceiver energy model
TPSN	Timing-Sync Protocol for Sensor Networks
WSN	Wireless sensor networks
WWAI	World wide article information



# LIST OF FIGURES

Figure 1 McBIM communicating concrete .....	2
Figure 2 Intelligent control system for communicating concrete.....	3
Figure 3 Information flow of closed-loop PLM (Jun et al., 2007).....	9
Figure 4 Classification of intelligent products (Meyer et al., 2009) .....	11
Figure 5 Intelligence classification with lifetime phases .....	12
Figure 6 Applying Intelligent product to PLM .....	13
Figure 7 Architectures of information system: EPCIS, DIALOG and WWAI.....	14
Figure 8 PROSA architecture (Van Brussel et al., 1998).....	15
Figure 9 Structure of an elementary material Holon .....	17
Figure 10 Representation of a communicating material .....	17
Figure 11 Example of WSN .....	18
Figure 12 Network architecture of communicating concrete.....	20
Figure 13 McBIM communicating concrete lifecycle .....	21
Figure 14 Intelligence levels of CM applications .....	22
Figure 15 Digital part design for communicating concrete.....	23
Figure 16 The holonic approach for communicating concrete.....	24
Figure 17 Architecture of sensor node .....	28
Figure 18 Classification of energy efficient mechanisms for WSN (Rault, 2017) .....	29
Figure 19 MLDA problem in WSN .....	30
Figure 20 Data aggregation example.....	30
Figure 21 The packet formats in application OscilloscopeRF (Jun-Zhao and Jiehan, 2008) ..	31
Figure 22 Tradeoff of data reduction functions.....	32
Figure 23 Hierarchical data collection structures in WSN.....	33
Figure 24 LEACH routing protocol .....	33
Figure 25 Minimum spanning tree-based routing protocol.....	37
Figure 26 PEGASIS chain-based protocol.....	38
Figure 27 CRBCC protocol.....	39
Figure 28 Timeline for chain based routing protocols (Khedr et al., 2021).....	40
Figure 29 RBS method .....	43
Figure 30 Synchronization between two nodes in TPSN.....	44
Figure 31 Radio energy dissipation model.....	45
Figure 32 Event trigger mechanism of NEM .....	46

Figure 33 MSP430 and CC420 current consumption information (Dron et al., 2014).....	47
Figure 34 PROSA architecture (Van Brussel et al., 1998).....	49
Figure 35 ADACOR architecture (Leitão and Restivo, 2006).....	51
Figure 36 Organization approach of agents: Flat (a), Hierarchical (b), Holonic (c), Coalition (d), Team (e), Matrix (f) and Congregation (g) (Dorri et al., 2018) .....	52
Figure 37 Holonic architectural model for embedded diagnosis .....	53
Figure 38 Social-based holonic architecture (Esmaeili et al., 2017).....	55
Figure 39 The MAS-R architecture.....	56
Figure 40 Three-level holonic architecture for urban planning (Khelifa and Laouar, 2020)...	57
Figure 41 Holonic architecture for microgrid (Nurdin and Muchallil, 2019).....	58
Figure 42 different states of radio module .....	63
Figure 43 Process at the furthest node.....	65
Figure 44 Process at intermediate node.....	66
Figure 45 Chronogram of radio module's activities without aggregation .....	67
Figure 46 Chronogram of radio module's activities with aggregation .....	68
Figure 47 Explanation of model parameters .....	69
Figure 48 Estimation of active duration for chain-based approach.....	72
Figure 49 WSN energy consumption method's overview .....	73
Figure 50 Aggregation coefficients at node .....	74
Figure 51 Data collection example with lossy aggregation .....	75
Figure 52 Transmitting duration for different size of data packet .....	76
Figure 53 Data processing flow of the message from node j.....	77
Figure 54 Root node is an intermediate node during data collection.....	79
Figure 55 Gathering two chains into a tree structure .....	80
Figure 56 Periodical data collection in tree structure.....	80
Figure 57 Tree as a composition of chains.....	81
Figure 58 Data collection in different structures.....	82
Figure 59 Radio interface consumption model for a node (one collection cycle) .....	84
Figure 60 Composed CM agent.....	88
Figure 61 CM recursive architecture.....	89
Figure 62 Agent model in MAS-R.....	90
Figure 63 Building recursive architecture with composition process .....	93
Figure 64 The sequence diagram of composition .....	94
Figure 65 Decomposition for aggregated CM.....	95

Figure 66 CM lifetime estimation process .....	97
Figure 67 Cooperation of two elements .....	98
Figure 68 Cooperation process for a new element .....	98
Figure 69 The first cooperation case: independent .....	99
Figure 70 The second cooperation case: association.....	99
Figure 71 The third cooperation case: aggregation .....	100
Figure 72 Data collection in independent concrete .....	101
Figure 73 Composition process for the construction of relationship matrix .....	102
Figure 74 Aggregation of two elements .....	105
Figure 75 Chain-based data collection .....	110
Figure 76 The materials of data collection platform .....	111
Figure 77 USB wattmeter.....	111
Figure 78 Radio module sleep mode test .....	112
Figure 79 Three-node chain structure .....	113
Figure 80 Estimated consumption of the whole chain .....	114
Figure 81 Active duration of nodes during data collection without aggregation.....	114
Figure 82 Six-node chain testbed .....	116
Figure 83 Test of 6-node chain structure.....	116
Figure 84 NetLogo McBIM energy simulator user Interface.....	118
Figure 85 NetLogo McBIM energy simulator 3D view.....	118
Figure 86 Different structures in NetLogo MAS simulator .....	119
Figure 87 Virtual network and multi-agent architecture in NetLogo.....	121
Figure 88 Energy consumption information obtained by NetLogo .....	122
Figure 89 Active duration of leader for data collection with lossless aggregation .....	123
Figure 90 Active duration of leader in data collection with lossy aggregation.....	123
Figure 91 FND time for data collection with lossy aggregation .....	124
Figure 92 Two independent communicating concretes.....	125
Figure 93 Cooperation between two communicating material elements .....	126
Figure 94 Cooperation between the aggregated agents.....	126
Figure 95 Composition of two communicating concrete .....	127
Figure 96 Reconstructed network.....	127
Figure 97 Energy consumption of nodes per round .....	128
Figure 98 Node lifetime for lossless aggregation.....	129
Figure 99 Node lifetime for lossy aggregation.....	129

# LIST OF TABLES

Table I. Summary of data aggregation protocols .....	41
Table II. Summary table of recursive MAS .....	59
Table III. Proposed model's symbols .....	64
Table IV. a. Knowledge part K concerning product-related information .....	92
Table IV. b. Knowledge part K concerning energy consumption in different abstraction levels .....	92
Table V. Parameter assumptions .....	101
Table VI. Estimation errors of activity durations in six-node chain structure .....	117

# Table of Contents

<b>ACKNOWLEDGEMENTS</b> .....	<b>I</b>
<b>ABSTRACT</b> .....	<b>III</b>
<b>RÉSUMÉ</b> .....	<b>V</b>
<b>LIST OF ACRONYMS</b> .....	<b>VII</b>
<b>LIST OF FIGURES</b> .....	<b>X</b>
<b>LIST OF TABLES</b> .....	<b>XIII</b>
<b>TABLE OF CONTENTS</b> .....	<b>XIV</b>
<b>INTRODUCTION: CONTEXT AND OBJECTIVES</b> .....	<b>1</b>
<b>CHAPTER 1 COMMUNICATING MATERIAL: CONCEPT, CONTEXT AND POSITIONING</b> .....	<b>8</b>
<b>1.1 Product Lifecycle Management</b> .....	<b>9</b>
<b>1.2 Intelligent Product</b> .....	<b>10</b>
1.2.1 Definition of intelligent product.....	10
1.2.2 Holonic paradigm .....	14
<b>1.3 A new type of intelligent product: the communicating material</b> .....	<b>15</b>
1.3.1 Definition of the communicating material .....	15
1.3.2 Holonic architecture for communicating material.....	17
1.3.3 Overview of Wireless Sensor Networks (WSN).....	18
1.3.4 Multi-Agent System .....	19
<b>1.4 CM application in the ANR McBIM project</b> .....	<b>20</b>
1.4.1 The communicating concrete design .....	20
1.4.2 Position of communicating concrete in the CM paradigm.....	22
<b>1.5 Proposed Holonic Approach and related research issues</b> .....	<b>23</b>
<b>CHAPTER 2 THE STATE OF THE ART ON PHYSICAL/DIGITAL PART</b> .....	<b>27</b>
<b>2.1 Physical part: Energy saving and estimation in WSN</b> .....	<b>28</b>
2.1.1 In-network data reduction .....	30
2.1.2 Data aggregation energy-efficient routing protocols.....	32
2.1.3 Time synchronization .....	42
2.1.4 Energy consumption models for WSN.....	45
2.1.5 Synthesis.....	47
<b>2.2 Digital part: energy management for communicating concrete</b> .....	<b>49</b>
2.2.1 Holonic Control Architecture (HCA).....	49

2.2.2 Multi-Agent System (MAS) .....	51
2.2.3 Recursive holarchy or MAS .....	53
<b>2.3 Synthesis .....</b>	<b>59</b>
<b>CHAPTER 3 ENERGY CONSUMPTION FOR PERIODIC DATA COLLECTION IN WSN.....</b>	<b>62</b>
<b>3.1 Energy model of a communicating node .....</b>	<b>62</b>
<b>3.2 Data collection in a chain communication structure.....</b>	<b>64</b>
3.2.1 Analytical model for data collection without aggregation .....	68
3.2.2 Analytical model for Data collection with aggregation .....	71
<b>3.3 Analytical model for data aggregation in WSN.....</b>	<b>72</b>
3.3.1 Energy consumption estimation method in WSN .....	72
3.3.2 Model for tree structure .....	79
3.3.3 Model for other structures .....	82
<b>3.4 Conclusion.....</b>	<b>83</b>
<b>CHAPTER 4 RECURSIVE ARCHITECTURE FOR ENERGY MANAGEMENT OF COMMUNICATING MATERIAL .....</b>	<b>87</b>
<b>4.1 The proposed CM recursive architecture .....</b>	<b>87</b>
<b>4.2 Adapted MAS-R agent model .....</b>	<b>89</b>
<b>4.3 Composition / decomposition mechanisms.....</b>	<b>93</b>
<b>4.4 Interaction between agents for lifetime estimation .....</b>	<b>96</b>
<b>4.5 Cooperation between two aggregated elements.....</b>	<b>97</b>
<b>4.6 Energy consumption estimation scenarios .....</b>	<b>100</b>
4.6.1 Data collection with lossy data aggregation .....	101
4.6.2 Energy consumption estimation for the composition of elements .....	104
<b>4.7 Conclusion.....</b>	<b>107</b>
<b>CHAPTER 5 EXPERIMENTS, DEVELOPMENTS AND VALIDATION.....</b>	<b>109</b>
<b>5.1 Validation of the energy consumption model for a chain structure .....</b>	<b>109</b>
5.1.1 Chain-based data collection platform.....	110
5.1.2 Energy consumption of Arduino nodes with XBee shield .....	111
5.1.3 Data collection in 3-node chain .....	113
5.1.4 Data collection in 6-node chain .....	115
<b>5.2 Validation of recursive architecture .....</b>	<b>117</b>
5.2.1 Analysis of energy consumption within a single communicating concrete .....	119
5.2.2 Using the simulator to analyze the First Node Death time.....	122

5.2.3 Cooperation of communicating concretes .....	125
<b>5.3 Analysis.....</b>	<b>130</b>
<b>GENERAL CONCLUSION AND FUTURE WORKS.....</b>	<b>132</b>
<b>REFERENCES .....</b>	<b>137</b>
<b>RÉSUMÉ DE LA THÈSE EN FRANÇAIS .....</b>	<b>146</b>



## **Introduction: context and objectives**

The rapid development of Internet of Things (IoT) leads to the intelligent product (IP) paradigm (Meyer et al., 2009). With equipped sensors, communication modules and computing chips, intelligent product has data storage and processing abilities. Product-related information is accessible to user directly via the RFID (Radio frequency identification), NFC (Near Fielded Communication) or remotely via other wireless technologies and the internet. The use of intelligent sensors allows to monitor surrounding environment or complete special tasks. Due to the accessibility of data and computing capacities, the intelligent product paradigm has been widely applied for Manufacturing, Logistics or Building Health Monitoring (McFarlane *et al.*, 2002) (Van Belle *et al.*, 2011) (Lamonaca *et al.*, 2018). However, there are some limits for the use of intelligent product: First, the information flow is interrupted if the equipped device is damaged; Second, the storage capacity within a product is limited and it is hard to ensure all product-related information throughout its whole lifecycle can be stored locally. To overcome these limits, this thesis considers the communicating material paradigm.

In 2009, the Research Center for Automatic Control (CRAN) began to study the “communicating material” (CM) concept. It is a material that can process, store data and communicate with the environment (Kubler *et al.*, 2010). CM is a paradigm which integrates the communicating ability into the material instead of equipping an external communication device. CM can sense its surrounding environment, store its monitoring data and its product-related information. Besides, a CM can either cooperate with others to form a composed CM or be decomposed into different parts, the composed or decomposed CM shall still be communicating materials. The concept has been applied in different works, such as (Kubler *et al.*, 2012) (Krommenacker *et al.*, 2013) (Jover *et al.*, 2013) (Mekki, 2016). These latter works have been proposed mainly for the traceability applications of textiles, or the data dissemination / data replication in concretes throughout their lifecycle. These works focus on communication ability.

This thesis conducted in the context of McBIM project (Material communicating with the BIM – Building Information Modelling) (ANR Project *McBIM*, 2017). Based on the communicating material concept, McBIM project aims to design a “communicating concrete” which is equipped with an embedded wireless sensor network (WSN). This project is funded by the ANR (French National Research Agency) and coordinated by the CRAN (Research Center for

Automatic control of Nancy) since the 1st of January 2018. The consortium is composed of four partners: the CRAN, two others research units (LAAS: the laboratory for Analysis and Architecture of Systems; LIB: Computer Science Laboratory of Bourgogne) and one company (360 Smart Connect/FINAO SAS).

McBIM project has two main objectives. The first is to design the communicating concrete which can monitor surrounding physical parameters, stores, process collected data and periodically transmit information to users (see Figure 1). The inner WSN consists of two types of nodes: sensing node (SN) and communicating node (CN). The SNs monitor or sense surrounding environment, and all interesting physical data (such as temperature, humidity or strength etc.) are transmitted to their dedicated CN. The CNs aggregate or process the received monitoring data and send the resulting message directly or indirectly (via other CNs) to a base station (BS).

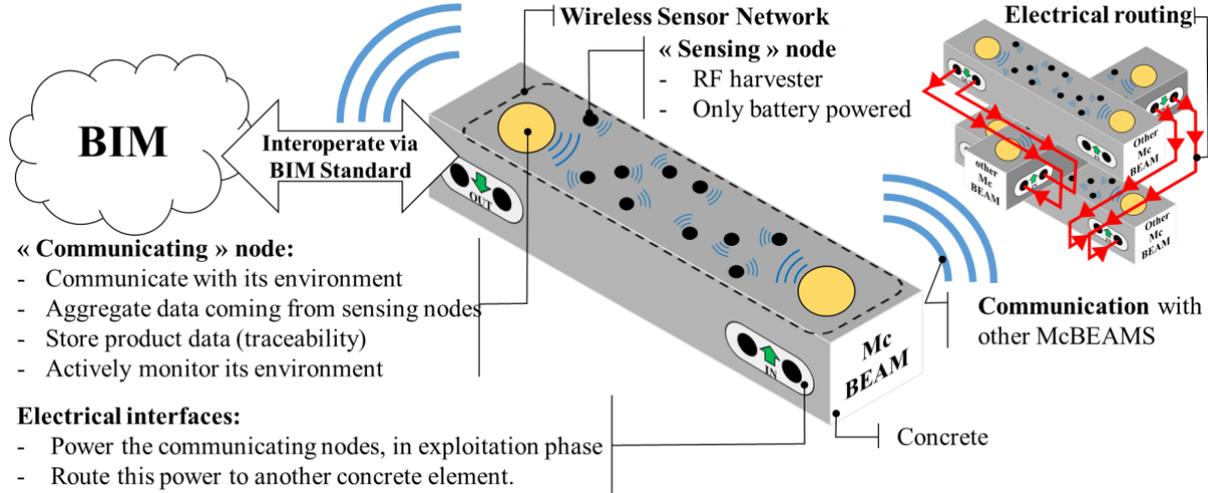


Figure 1 McBIM communicating concrete

The second objective is to demonstrate the usefulness of communicating concrete over whole building lifecycle. In different phases, McBIM concretes may be exploited or used by different companies with their own BIM data. As consequence, their monitoring behaviors may be also different (for example the monitoring frequency). Long-term service requirement, the need of large data storage capability and high flexibility have brought different challenges.

The different partners collaborate to achieve these objectives. The LAAS designs the sensing and communicating nodes, the LIB studies data interoperability with BIM applications. 360 Smart Connect/FINAO implements the result of research partners in real application. The CRAN works on the intermediate layers between the hardware (developed by LAAS) and the

application (developed by LIB and 360SC). In the framework of this project, **my thesis focuses on the energy management for periodical data collection realized by the embedded WSN.**

Since the communicating concrete should work at least ten or twenty years, one element may cooperate with the others to save energy. Due to the limitation of storage and computational capability of the communicating concrete, this work explores the coupling of this physical artefact with a numerical representation, in the form of a multi-agent system, to manage data and energy of embedded network (see Figure 2). Indeed, communicating nodes in the physical part can be represented by dedicated agents in digital part which can not only store the monitoring and product-related information, but also evaluate and estimate data collection strategies to provide an optimal solution with a global view. This digital part of communicating concrete can be seen as a digital twin.

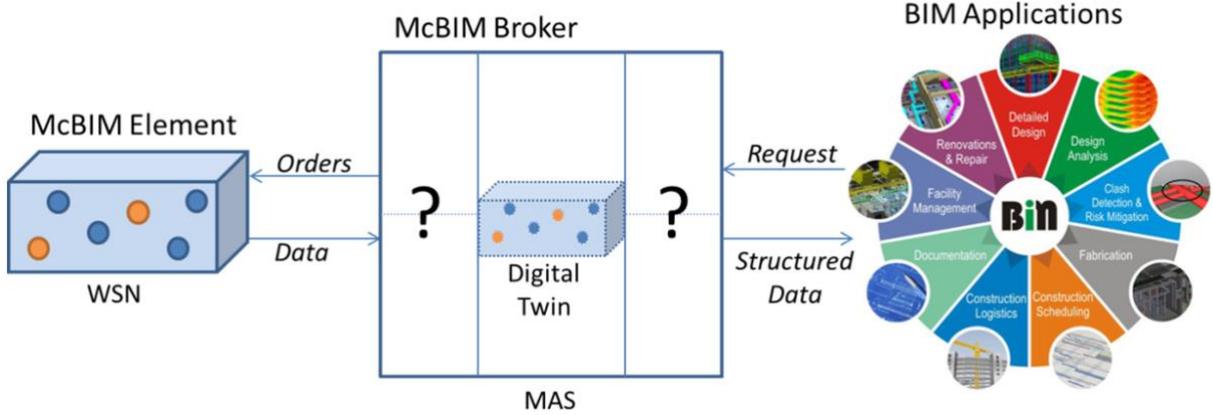


Figure 2 Intelligent control system for communicating concrete

To achieve an efficient energy management and extend network lifetime, there are four challenges to tackle: two for the physical part (questions 1 and 2, related to the embedded WSN), and two associated to the digital part (questions 3 and 4, concerning data and energy management) as follows:

1. *How to maximize lifetime of data collection (process) within the embedded WSN?* Both data processing techniques, duty cycle management at node and data collection strategies between nodes should be taken into account to reduce energy cost for data collection.
2. *How to predict remaining energy of network in a resource constrained environment?* To predict the remaining energy, consumption models must be established. Is it possible to build a generic model or is it necessary to specify energy models for each kind of WSN structures?

3. *How to realize the link between the communicating concretes with their dedicated agents, as well as data and energy management of nodes at these agents?* The physical part is linked to a digital part represented in this work by a multi-agent system. As a result, this issue studies the relation between real physical nodes and the constitution of the agent architecture in the MAS.
4. *How to achieve the composition of agents?* Communicating material is a product which has to support composition/decomposition features. This could be possible at the physical level and at the digital level at the same time. To build this mechanism, the interactions between agents should be considered.

Around these mentioned challenges, the main contributions of this study, both theoretical and practical are the followings:

- Analytical energy consumption models for periodic data collection in WSN are proposed. An energy model for data collection is detailed, as well as the model for data collection without aggregation (see *Chapter 3*). These predictive models are tested and validated by experimentations (see *section 5.1 of Chapter 5*).
- A recursive MAS architecture is presented (*Chapter 4*) to build a digital part for communicating material. The proposed architecture is recursive and flexible to allow communicating materials composition/decomposition. The digital part allows to provide the remaining energy level but also to provide cooperative solutions to meet high flexibility of communicating concrete.
- A 3D communicating concrete simulator is developed with the NetLogo platform, where communicating nodes are represented by agents. This digital tool allows to visualize the energy level of communicating concrete and provides the remaining lifetime information (in *section 5.2 of Chapter 5*).

This thesis is organized in five chapters followed by a general conclusion:

*Chapter 1* introduces the context and the concepts of communicating material, its characteristics and interests in the framework of the McBIM project. It begins with the context of Product Lifecycle Management. IP and the holonic paradigm are then described. To overcome the limitations of intelligent product, the concept of communicating material (CM) is proposed. In order to efficiently manage and extend the lifecycle of communicating concrete, the solutions

in WSN are discussed, meanwhile a holonic architecture is proposed for data and energy management. Finally, the research issues of this thesis are detailed.

*Chapter 2* presents a state of the art both on the physical part and the digital part. For the physical part, literature solutions for in-network data processing in WSN are presented: data aggregation functions, energy efficient routing protocols, synchronization protocols and energy consumption models. For the digital part, the traditional holonic architecture for intelligent product is presented, as well as the multi-agent approaches for information management. In addition, to meet the composition/decomposition requirement of CM application, recursive solutions are reviewed and discussed.

*Chapter 3*, related to the literature on physical part, presents a power-based energy consumption model for communicating node. Based on this model, analytical energy models are proposed to estimate energy consumption for data collection with/without aggregation in WSN. In addition, analytical energy models for lossy or lossless data aggregation collection are also discussed for any kind of WSN topology structures.

*Chapter 4*, related to the literature on digital part, details the proposed Multi-Agent System (MAS) recursive architecture and the communicating material agent model. Besides, composition/decomposition mechanisms are discussed, information handling and sharing between agents are detailed especially for energy management as well as the use for lifetime estimation and composition/decomposition.

*Chapter 5* presents the applicative and demonstrative results for both the physical (WSN) and digital part (MAS). The analytical energy models for data collection with/without aggregation are validated with an Arduino-XBee platform (in *section 5.1*). On the digital part, a 3D agent-based model is developed on the Netlogo platform for visualizing energy level of communicating node, analyzing and providing optimal solution to extend lifetime of the physical part.

Finally, the thesis ends with a general conclusion in which the contributions of our works are summarized, and some perspectives are discussed.

The structure of this thesis is presented as follows:

### **Introduction: context and objectives**

This part describes the context motivation of this thesis, the objectives as well as facing research challenges and our main contributions (*current part of the document*).

### **Chapter 1: The concept of communicating material**

This chapter introduces the concept of communicating material, its characteristics, and its application interests in the framework of the McBIM project. Besides, a holonic architecture is presented for the energy management.

### **Chapter 2: The state of the art on physical part and digital part**

Chapter 2 presents the state of the art both on the physical part (embedded WSN) and the digital part (Multi-Agent system).

### **Chapter 3: Energy consumption for periodic data collection in WSN**

Chapter 3 presents the energy consumption model of communicating node and data collection with/without aggregation in chain structures are detailed. Data collection with lossy or lossless aggregation are also discussed for chain, tree, and cluster structures.

### **Chapter 4: Recursive architecture for energy management of Communicating Material**

Chapter 4 presents the proposed recursive architecture and the communicating material agent model. The composition/decomposition mechanisms are discussed, followed by the cooperation between aggregated CMs.

### **Chapter 5: Experiments, developments and validation**

The relative experiments for data collection with/without aggregation on an Arduino-XBee platform are illustrated in this chapter. Meanwhile, a 3D-energy estimation on Netlogo platform for digital part is presented for the remaining lifetime estimation as well as the cooperation between CMs.

### **Conclusion**

This chapter summarizes the contributions of this work and points out the following perspectives.



# Chapter 1

## Communicating Material: concept, context and positioning

### Contents

<a href="#"><u>1.1 Product Lifecycle Management</u></a> .....	9
<a href="#"><u>1.2 Intelligent Product</u></a> .....	10
<a href="#"><u>1.2.1 Definition of intelligent product</u></a> .....	10
<a href="#"><u>1.2.2 Holonic paradigm</u></a> .....	14
<a href="#"><u>1.3 A new type of intelligent product: the communicating material</u></a> .....	15
<a href="#"><u>1.3.1 Definition of the communicating material</u></a> .....	15
<a href="#"><u>1.3.2 Holonic architecture for communicating material</u></a> .....	17
<a href="#"><u>1.3.3 Overview of Wireless Sensor Networks (WSN)</u></a> .....	18
<a href="#"><u>1.3.4 Multi-Agent System</u></a> .....	19
<a href="#"><u>1.4 CM application in the ANR McBIM project</u></a> .....	20
<a href="#"><u>1.4.1 The communicating concrete design</u></a> .....	20
<a href="#"><u>1.4.2 Position of communicating concrete in the CM paradigm</u></a> .....	22
<a href="#"><u>1.5 Proposed Holonic Approach and related research issues</u></a> .....	23

This chapter aims at introducing the concept of communicating material (CM), its application interests and our study case called McBIM project. As explained later, CM is a sub-type of *intelligent product*, concept used in diverse applications (production control, maintenance, ...) in past years. One important application field of this concept is the management of product data all over its lifecycle, which is also our application context. This chapter begins with a description of product lifecycle management (*section 1.1*), followed by a short presentation of the notion of intelligent products in *section 1.2.1*. In order to effectively manage the intelligent products and their information, the *Holonic Paradigm* is presented in *section 1.2.2*. The definition and characteristics of communicating material is given in *section 1.3*, followed by the specific application case of the ANR McBIM project in *section 1.4*. Finally, the proposed digital twin design for data and energy management as well as the related research issues are given in *section 1.5*.

## 1.1 Product Lifecycle Management

In the context of globalization, product lifecycle management (PLM) plays an increasingly important role in enterprises. PLM emerged in the 1990s but has been conceptualized by (Rachuri et al., 2008) as a strategic approach to create and manage a company's product-related intellectual capital, from the design to the retirement of product. Since a product may be used by many different companies throughout its whole lifecycle, large amounts of information are stored, exchanged and managed.

In general, the product lifecycle consists of three main phases: 1) Beginning of Life (BOL), including the product design and manufacturing; 2) Middle of Life (MOL), including logistics, exploitation and maintenance; 3) End of Life (EOL), including remanufacturing, reuse and recycling. In different phases, the product may be used by different stakeholders or companies with their own product management systems or standards. The difference of management standards or systems can create "information silos" and may cause interruptions in product information flow. To prevent users to make decision with incomplete information, the concept of closed-loop PLM has been proposed by (Jun et al., 2007). An example of information flow is shown on Figure 3, depicting direct and feedback (indirect) information flows among BOL, MOL and EOL.

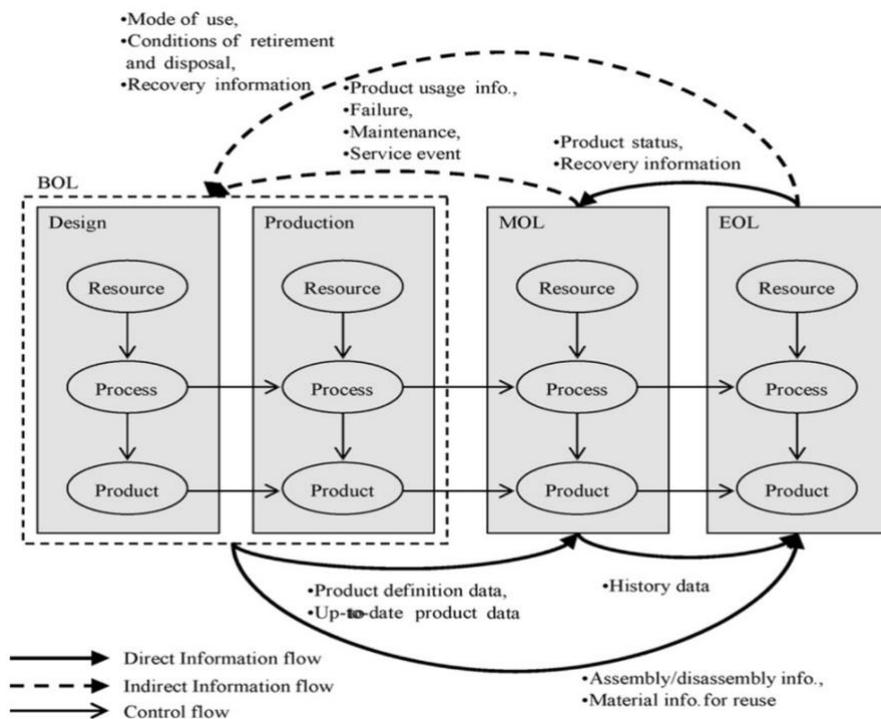


Figure 3 Information flow of closed-loop PLM (Jun et al., 2007)

Compared with traditional PLM approaches, the closed-loop PLM focuses on the product information management over the whole lifecycle. Thus, all actors (manufacturer, transporters, and customers) can make better operation decisions with the complete product information.

## 1.2 Intelligent Product

Substantial information distribution improves data accessibility and availability compared to centralized architectures. Product information may be allocated both within fixed databases and/or within the product itself, thus leading to products with informational and/or decisional abilities, referred to as “Intelligent Products”. Several different definitions of this concept have been proposed and are presented in the next section.

### 1.2.1 Definition of intelligent product

The concept is first proposed in (McFarlane et al., 2003), as a physical product that can support or influence operations to meet set conditions or purposes. It has five characteristics:

- Possesses a unique identity.
- Is capable of communicating effectively with its environment.
- Can retain or store data about itself.
- Deploys a language to display its features, production requirements, etc.
- Is capable of participating in or making decisions relevant to its own destiny.

According to Kärkkäinen, intelligent product is a product centric approach, which uses software agents, peer-to-peer information sharing, and coding mechanism enabling the material information flows (Kärkkäinen et al., 2003). The definition given in (Ventä, 2007) focuses on the communication and reaction capabilities of the product with its surrounding environment. According to Venta, intelligent products should have the four following characteristics: 1) Continuously monitor their status and environment; 2) React and adapt to environmental and operational conditions; 3) Maintain optimal performance in variable circumstances; 4) Actively communicate with the user, environment or with other products and systems. (Kiritsis, 2011) defines that an intelligent product is a product system which contains the three capabilities: 1) sensing and communicating with environment; 2) storing its own data; 3) processing monitoring or received data to react to environment changes.

Product intelligence is a wide field encompassing different research areas and applications, such as manufacturing control in (McFarlane *et al.*, 2002), supply chain in (Van Belle *et al.*, 2011) and structural health monitoring in (Lamonaca *et al.*, 2018). A three-dimension classification is proposed in (Meyer *et al.*, 2009) and illustrated Figure 4. It is composed of 3 different orthogonal axes: *Intelligence level*, *Location of intelligence*, and *Aggregation level of intelligence*.

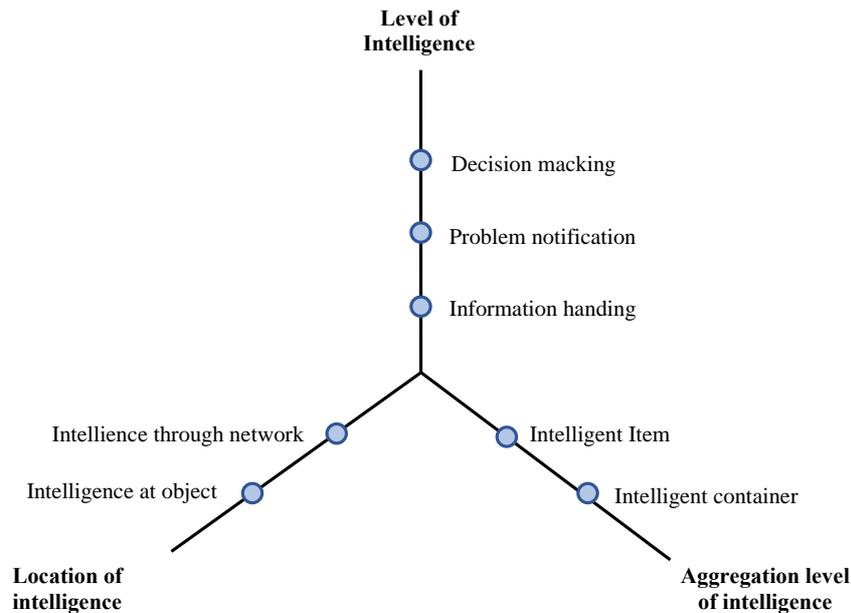


Figure 4 Classification of intelligent products (Meyer *et al.*, 2009)

On the *level of intelligence* axis, there are three sub-levels: *information handling*, *problem notification* and *decision making*. An intelligent product at the *information handling* level can only process and transmit the monitoring information. With the *problem notification* ability, the product can detect unwanted situations and report warning signals. At the highest level, the product can adjust its own state and can adapt to environmental changes.

There are two possibilities under the intelligent location axis: *intelligence through network* and *intelligence at object*. In the first one, the intelligence of the product is outside the product, namely, on an external server that processes and generates operation orders to react to environment changes. In the second one, all information processing, even advanced decision, is done by the product itself.

There are two levels corresponding to the aggregation dimension: *intelligent item* or *intelligent container*. The *intelligent item* only manages its own information, notifications and decisions.

If it contains components, they cannot be distinguished as individual objects. In contrast, the *intelligent container* is aware of the states of all components. When parts of its composition are removed, it can still work as a smart container. For example, if the engine of a car is removed, the car can still know the states of the other components.

Another classification is proposed in (Zbib, 2011) which takes all lifecycle phases into account as shown in Figure 5. Compared with the classification proposed by Meyer, the *location of intelligence* and *aggregation of intelligence* axes are the same. There are two more levels in the degree of intelligence, and a new axis introduces the time dimension: Lifecycle.

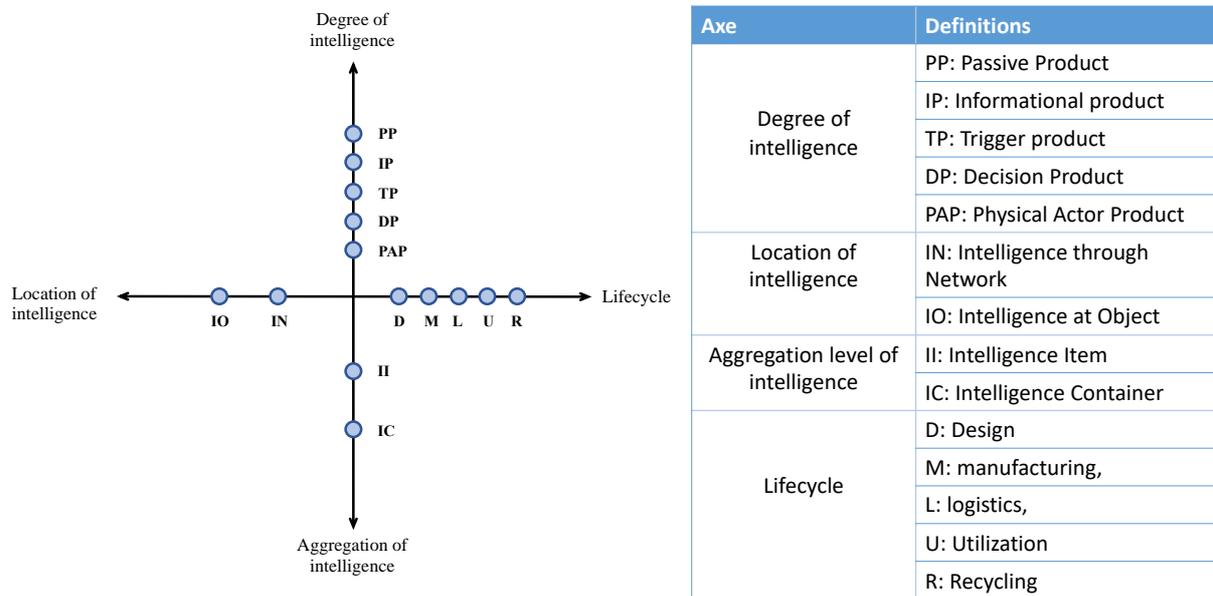


Figure 5 Intelligence classification with lifetime phases

On the degree of intelligence axis, there are two additional levels: Passive Product (PP) which can only store the product-related information, and the Physical Actor Product (PAP) which can react to the environment changes based on its decisions. On the lifecycle axis, all product phases (design, manufacturing, logistics, utilization, and recycling) are considered. Indeed, the intelligence of a product may be different in different phases. Both classifications can be used to analyze the intelligence requirement of an intelligent product in different lifecycle phases.

With the communication, processing and storage capabilities, intelligent product has been widely used for PLM from manufacturing to recycling as shown in Figure 6. A large number of product-related information may be generated throughout the whole lifecycle. To overcome the limitation of data storage and information processing, product-related information can be stored in remote servers, and dedicated agents can be used to complete decision-making

processing to adapt to environmental changes. As explained in the next section, an intelligent product with its dedicated decision-making agent can be seen as a product holon. As consequence, a holonic architecture could be considered for the information and energy management of intelligent product.

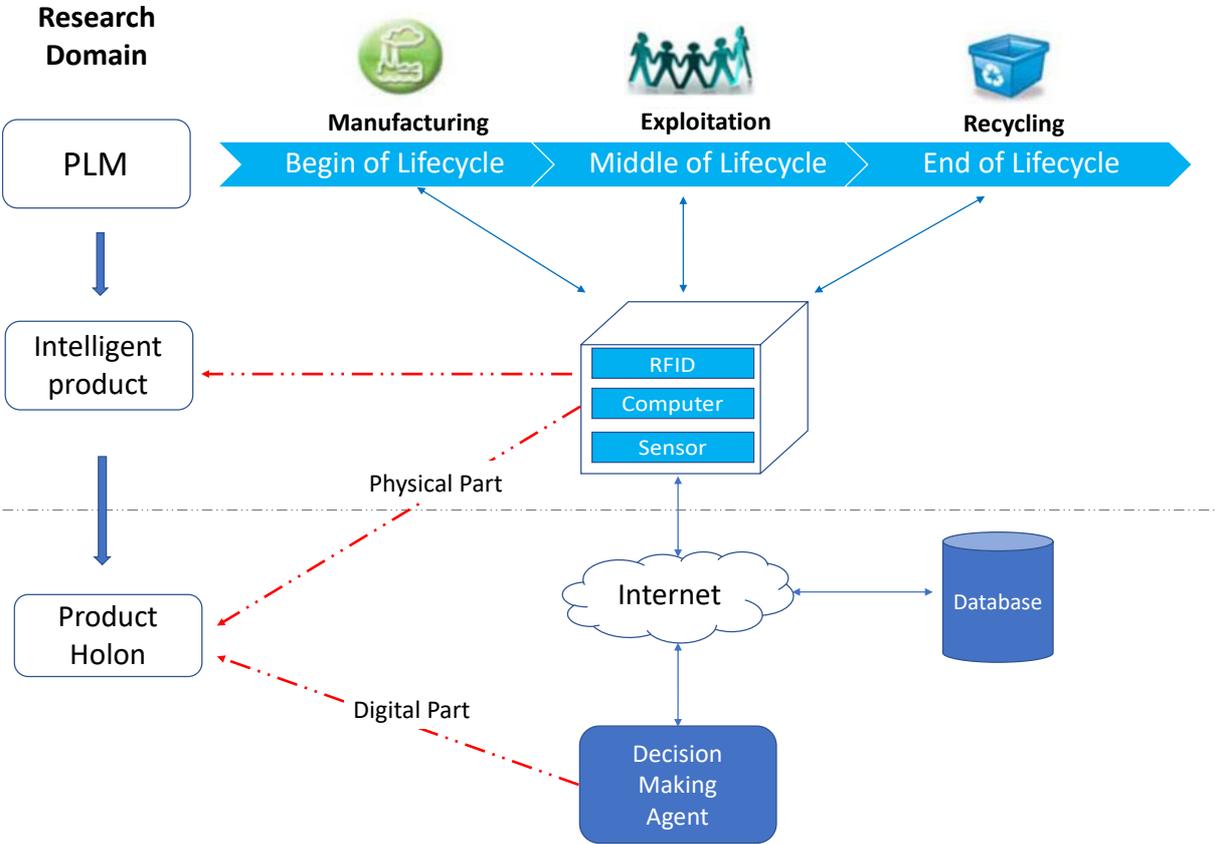


Figure 6 Applying Intelligent product to PLM

Some approaches have been proposed for the information management, such as the EPCglobal (EPCglobal, 2001), DIALOG (Dialog, 2003) and WWAI (World Wide Article Information, 1992).

The architecture of EPCglobal is shown in Figure 7. Each product is labeled with a unique electronic product code (EPC). The product information is accessed via the reader, and then filtered and sent to the application layer EPCIS (EPC Information Service). In this centralized approach, it is the ONS (Object Naming Service) that indicates where to find the required information. In DIALOG, the product identification and information linking concept ID@URI was used to create a globally unique product identifier, where the URI is a computer address (for example "www.xxx.com") and ID represent the serial number of product instance. Product-related information is managed by a product agent in server. The use of ID @ URI provides a

pointer that allows users to query the product agent for their required information. Unlike the two other standards, WWAI is a peer-to-peer application-level protocol for distributed article information. The product identifier is the WWAI identity code. In this approach, there is no central storage of product information. Each participant controls its own product information and decides whether the information is public or private.

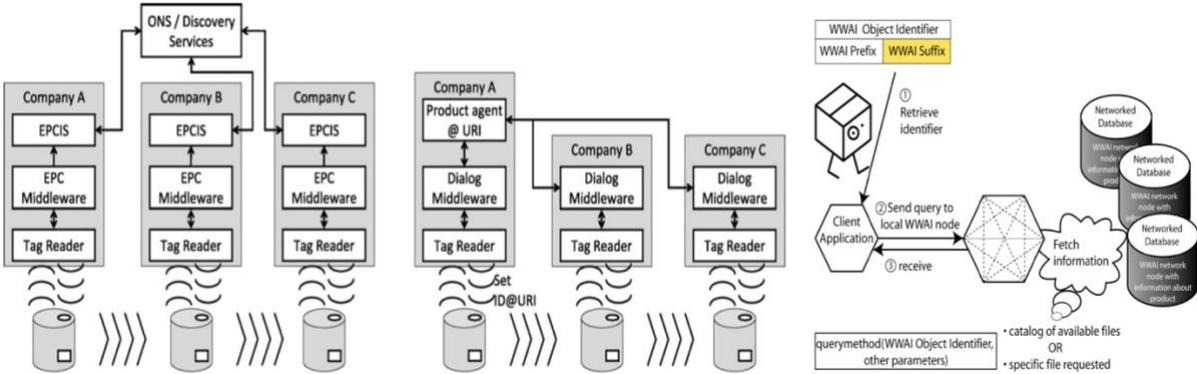


Figure 7 Architectures of information system: EPCIS, DIALOG and WWAI

Hereafter, a distinction is made between “the physical part” (the product augmented with communicating devices) and the “digital part” (software components stored on the cloud). The physical part combined with the digital part can be seen as a *holon*, which is presented in the following section.

### 1.2.2 Holonic paradigm

The concept of *holon* is first proposed by Koestler (Koestler, 1968). This term is a combination of the Greek word *holos*, meaning “whole”, with the suffix *-on* standing for “part”. A holon consists of a physical part and digital part. According to the Janus effect, holons can be grouped into a whole to form hierarchical structures called *holarchies*. According to the definition, a holon has two main characteristics:

- **Autonomy:** each holon can act autonomously, even in case of unpredictable circumstances.
- **Cooperation:** holons are able to negotiate and execute mutually acceptable plans and take mutual actions against malfunctions.

A holon can represent a physical or logical activity, like a machine or a control system. For example, in the holonic manufacturing architecture PROSA (Product-Resource-Order-Staff) (Van Brussel *et al.*, 1998), products, resources and orders are represented by different types of

holons as shown in Figure 8. Besides, staff holons are introduced to assist the basic holons to complete their work. In this architecture, these holons can cooperate together to achieve common goals.

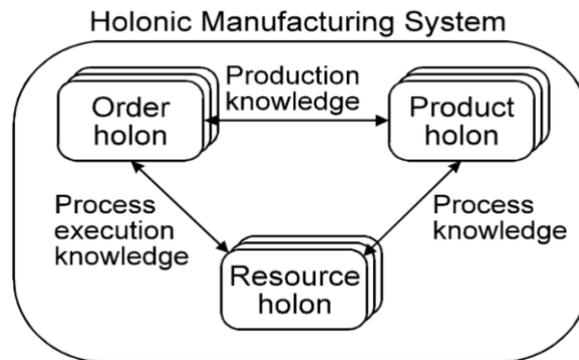


Figure 8 PROSA architecture (Van Brussel et al., 1998)

These highly flexible features make holonic paradigm having been widely used in manufacturing (Chirn and McFarlane, 2000) (Morariu et al., 2013) (Sallez et al., 2015). Multi-agent Systems are often used to represent the virtual part of a holon.

## 1.3 A new type of intelligent product: the communicating material

### 1.3.1 Definition of the communicating material

Thomas proposes the concept of *communicating materials*, i.e., materials able to communicate with their environment, process, exchange and store information in their own structures (Thomas, 2009). Besides, they also have the capability to sense their environment and measure their own internal physical states. In the works of (Kubler et al., 2012) and (Mekki, 2016), this concept has been applied in different industrial domains, from different perspectives. Diverse early prototypes were designed (or simulated) for the needs of the manufacturing and the construction industry, by spreading micro-electronics devices into a material.

The material could be either wood, textile or concrete. The interests of such material are various: (a) because of their data storing capacity, they can convey all information related to design, manufacturing and logistics throughout whole lifecycle, (b) given the ability to sense their environment and process related information, they can also be used during the MOL (exploitation and maintenance), such as intelligent building sensors to perform structural health monitoring. In previous works, the inserted devices were either RFID tags (Kubler et al., 2012)

or self-powered wireless sensor networks (WSNs) embedded into the material (Mekki et al., 2014). Both works deal with the data dissemination/data replication, which is an issue related to the second capability. However, these works did not address the issue related to the first capability, i.e., the composition/decomposition of communicating materials.

No clear formal definitions of the communicating material have been provided until now. As consequence, a definition is provided hereafter. In our work, the Communicating Material is an Intelligent Product with two additional capabilities:

- ***The capability of being intrinsically and wholly communicating:*** even if the product undergoes a physical transformation (composition or decomposition), the resulting pieces shall still be communicating materials.
- ***The capability of managing its own data:*** The material should be able to manage its own data according to the events occurring in its environment. For instance, the material could decide itself to propagate/replicate specific data onto different material parts because a physical transformation is scheduled, thus avoiding data losses.

Therefore, based on the definitions of intelligent products in (Wong et al., 2002) (Ventä, 2007), the communicating material has the following 7 characteristics:

1. Possesses a unique identity
2. Is capable of communicating effectively with its environment
3. Can retain or store data about itself
4. Deploys a language to display its features, production requirements, etc.
5. Is capable of participating in or making decisions relevant to its own destiny
6. Continuously monitors its status and environment
7. Can remain a communicating material under composition or decomposition.

Characteristics (1) to (5) are inherited from (Wong et al., 2002), Characteristic (6) is from (Ventä, 2007). This one is important since it underlines the capacity of a product to monitor its own status or environment. Characteristic (7) is the only one which is completely dedicated to the concept of communicating material and is mandatory.

### 1.3.2 Holonic architecture for communicating material

As a special type of intelligent product, communicating materials have composition and decomposition characteristics which is also a fundamental property of Holons as introduced by (Koestler, 1968). Therefore, a CM can be considered as a material holon which consists of communicating nodes in the physical part related to dedicated node agents in digital part as shown in Figure 9. The node agent can not only store the product-related information throughout the whole lifecycle, but also provide suitable strategies to real nodes to adapt to environment changes based on the state of nodes and on the monitoring information.

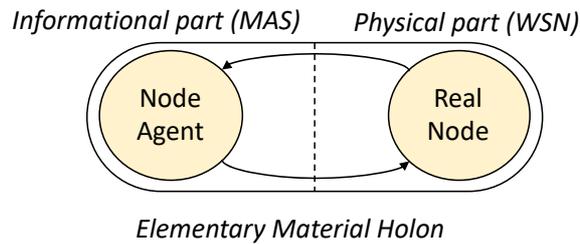
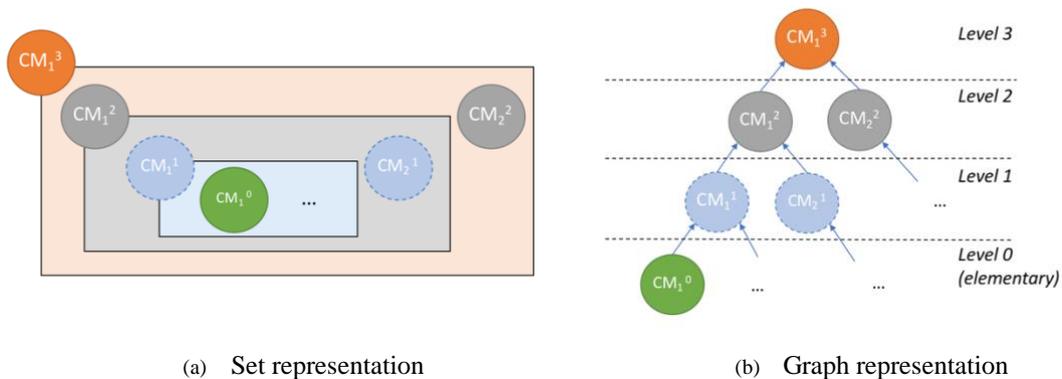


Figure 9 Structure of an elementary material Holon

As communicating material can be considered as a type of holon, we can then define a CM as a holarchy of sub-communicating material  $CM_j^i$ , where  $i$  is the level of holarchy the CM is attached to, and  $j$  the index of the CM in the corresponding level. Each sub-communicating material could be decomposed in other communicating materials until a primary level of decomposition called *elementary* depicted in Figure 9. Indeed, even if the level of the holarchy could be infinite, it is limited in practice. The elementary level is the finest granularity of the holarchy. Different representations of a communicating material are illustrated in Figure 10. As the composition can be infinite, a recursive system for the management of informational part is necessary.



(a) Set representation

(b) Graph representation

Figure 10 Representation of a communicating material

A communicating material should work as long as possible, however, there are physical limits, especially concerning the energy. Therefore, energy management solutions in the physical part (WSN in our work) and the digital part (MAS in our work) should be both considered for extending its lifecycle. The next sections will briefly introduce the two components of the communicating material holon, i.e. the WSN (*section 1.3.3*) and the MAS (*section 1.3.4*).

### 1.3.3 Overview of Wireless Sensor Networks (WSN)

The rapid development of microprocessors and sensors has made WSN widely used in various fields (Ansola et al., 2012), such as environment monitoring (Paek et al., 2014), smart agriculture (Li et al., 2011), building monitoring (Bhuiyan et al., 2015) and logistics (Xiao et al., 2016). Most applications are in resource constrained environments. The limitation of energy brings challenges related to the network lifetime and performance. In our study, since it is hard to change the battery for sensor node in concrete, we focus on maximizing the lifetime using data aggregation (MLDA) in WSN. Existing solutions will be presented in the *section 2.1*.

Wireless Sensor Networks are often made up of a large number of nodes as shown in Figure 11. Sensor nodes have the capabilities of sensing, processing and communicating data. They monitor their surrounding environment (such as temperature, humidity, pressure, motion, or other physical phenomena) and transfer their data directly or indirectly (through other nodes) to a base station (BS) also named sink node.

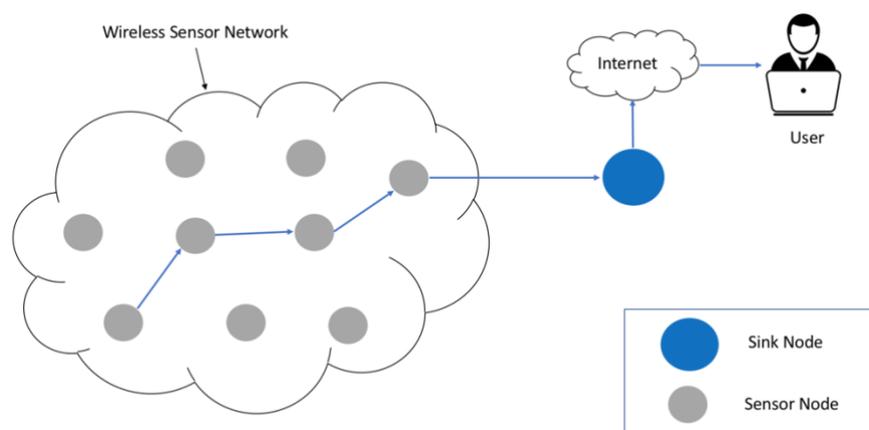


Figure 11 Example of WSN

Sensor nodes send monitoring data to their neighbor up to the sink node (in blue) which then transfers data over the internet. Over last two decades, how to extend the network lifetime has

attracted the attention of many researchers (Kalpakis and Tang, 2009) (Aziz et al., 2013) (Yetgin et al., 2017) (Wang and Pan, 2020). Due to the limitation of technologies, the processing power and storage capability of nodes is limited. Nodes can not always have a global view of the network, thus optimization over the network is often suboptimal. As consequence, applying an efficient external management system with a global view of the network could provide suitable solution to extend network lifetime.

### 1.3.4 Multi-Agent System

As mentioned above, to represent the physical part in holarchies, multi-agent systems are often implemented, since they can solve complex tasks (Sardouk et al., 2013) (Dorri et al., 2018). A MAS is composed of multiple autonomous agents that are capable to achieve their individual goal (Giret and Botti, 2003). The agent is a virtual entity which can process monitoring information and make decisions based on a given goal. According to Wooldridge (Wooldridge and Jennings, 1995), an agent has four characteristics:

- **Autonomy:** the agent can work alone without human intervention.
- **Social ability:** the agent can interact with other agents via a communication language.
- **Reactivity:** the agent senses its environment and responds in a timely manner to modify the course of events if necessary.
- **Pro-activeness:** each agent uses its history, monitored data and information of other agents to predict future actions.

Due to its autonomy and high flexibility, it has been widely used from industry to our daily life. However, to face the non-deterministic environment, two major questions are pointed out by Wooldridge: 1) how to build autonomous agents which can make their actions independent to complete delegated tasks even with environment changes; 2) how to define relationship for agents with common goals or different goals (Wooldridge, 2009).

Different authors have attempted to answer and validate these two questions. According to Caridi, optimal global performance could be achieved if several decision-making agents cooperate for the common interest or task (Caridi, 2004). From Isern, even if the agents are perceived as autonomous entities, the societal relation is necessary to allow the communication between the agents to achieve a global optimization (Isern et al., 2011). Agents can communicate with each other via some defined architecture to achieve common goal.

In the following section, the CM application in McBIM project will be presented. The interests of communicating concretes for PLM will be discussed as well as its position in the CM paradigm.

### 1.4 CM application in the ANR McBIM project

#### 1.4.1 The communicating concrete design

Based on the concept of CM, the ANR McBIM (Material communicating with the BIM – Building Information Model) project aims to design a “communicating concrete” which is equipped with an embedded WSN. The network architecture of communicating concretes is shown in Figure 12. There are two types of nodes: the sensing node (SN) and the communicating node (CN). The SNs measure relevant parameters and transmit measured data to CNs. The CNs then process, store and transfer the received data to the base station.

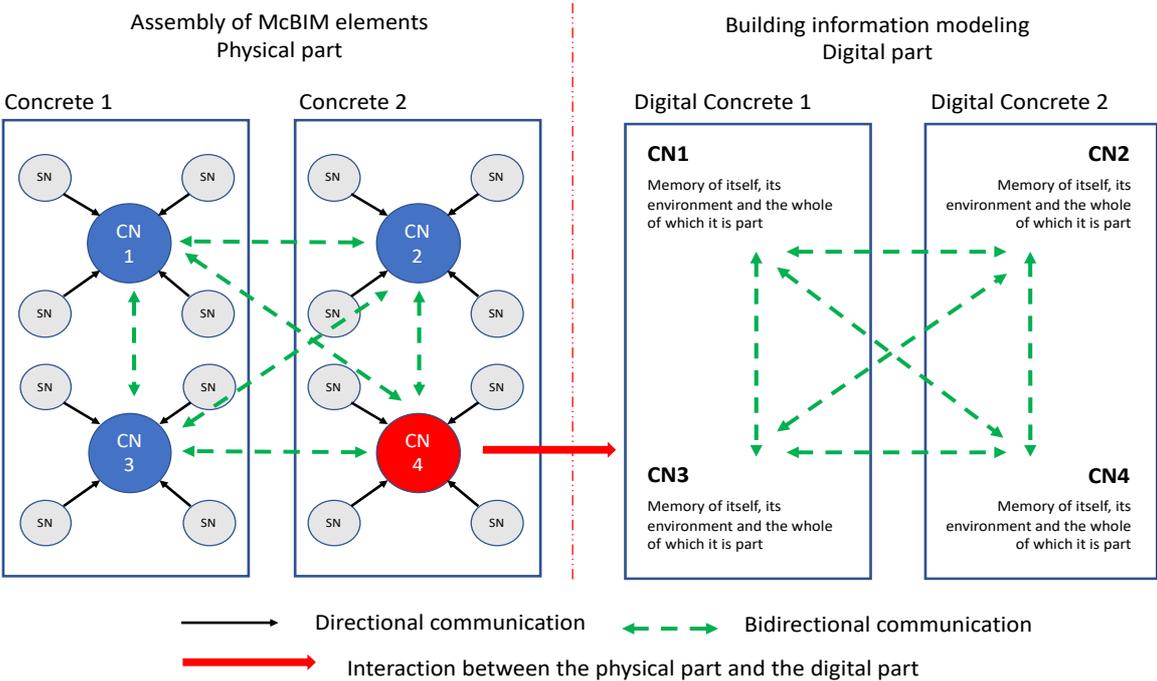


Figure 12 Network architecture of communicating concrete

Thanks to the embedded WSN, communicating concretes can monitor, process and send product-related information (like temperature, humidity, corrosion and so on) which is useful all along the concrete lifecycle as shown in Figure 13. During the manufacturing phase, the WSN nodes are inserted and initialized. Communicating concretes periodically (by example

every hour) monitor their physical status, store the physical propriety information, as well as the information of manufacturer, constructors and operators. These data are directly accessible via a reader device or remotely via the internet. The monitored humidity information can avoid excess precuring time and improve manufacturing efficiency. The logistics information can help manufacturers and transporters to reduce logistic and inventory costs.

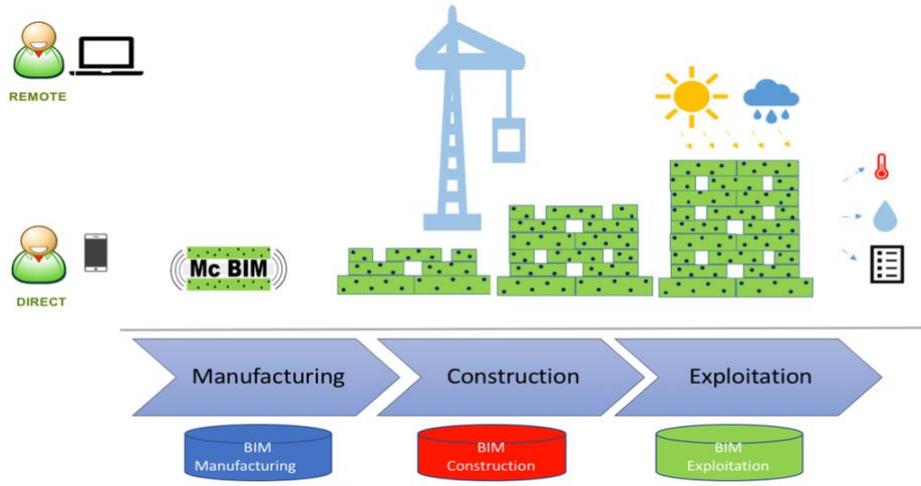


Figure 13 McBIM communicating concrete lifecycle

During the construction phase, communicating concretes will be assembled together. In this case, auto-organization is then needed to dynamically define a 3D network to achieve better data collection and thus energy savings. Due to the high flexibility in this step, concrete has to report its status frequently (such as every minute) to ensure the safety of construction and update the network information. When the construction is completed, the large 3D static WSN will regularly (by example once a day or once a week) monitor structural health data (such as cracks, temperature, corrosion, etc.) to ensure the safety of the building.

Since communicating concretes should work from manufacturing to exploitation (twenty or thirty years), long-term and high-quality service requirements bring challenges related to energy management in this inaccessible environment. Therefore, being able to monitor and/or determine the consumption of each node of the network is crucial for its management. Besides, a large number of product-related information will be generated throughout the whole lifecycle. Due to the limitation of data storage, it is hard to store all information in the concrete. Therefore, applying the concept of Digital Twin (DT) seems to be a suitable solution. It would bring a WSN virtual model to store and present all information concerning the physical part and provide optimal behavior decisions (with a centralized solution) to extend its lifetime.

### 1.4.2 Position of communicating concrete in the CM paradigm

Concerning the intelligence levels of communicating concrete application in McBIM project, the mentioned classifications in (Meyer et al., 2009) and (Zbib, 2011) could be used to position this application of the CM paradigm, that can be compared with the previous works of (Kubler, 2012) and (Mekki, 2016). Although the behavior of communicating concrete may be different throughout the whole lifecycle, its intelligence on the axis of information processing level, intelligence location and aggregation level stay the same all along its lifecycle. Therefore, the classification of (Meyer et al., 2009) is used to position CM works as depicted in Figure 14.

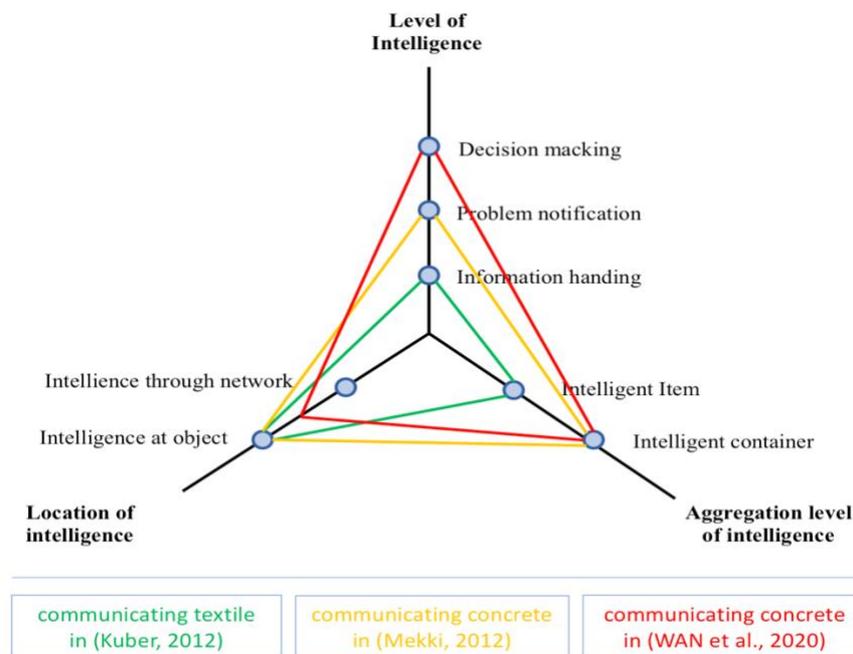


Figure 14 Intelligence levels of CM applications

For the axis *level of intelligence*, communicating textile can only store product-related information (*Information handing* level). Thanks to the computing ability of sensor nodes, the communicating concrete developed in (Mekki, 2016) equipped with an embedded WSN can send warning messages to the server (for example, a too high temperature) if the monitoring data exceeds the set value (*Problem notification* level). As mentioned above, communicating concrete in McBIM project can make decision locally or by their dedicated agent to extend product lifecycle. Therefore, the level of intelligence for CM application in McBIM is the decision-making level.

Mekki proposes network algorithms for data collection, based on routing choices made locally, by each node. Even if it can be locally optimal, the solution is globally suboptimal. In the

McBIM Project, the MAS can provide an optimal data collection strategy because it owns a global view of the embedded network. This global strategy could locally be mitigated by the physical node view. This explains why the location of intelligence is between intelligence at object and intelligence through network.

Concerning the aggregation level dimension, composition/decomposition is not considered in the application of communicating textile. For the communicating concrete application in (Mekki, 2016), decomposition has been discussed for data storage and dissemination. In the McBIM project, communicating concrete can work independently by example to report its states during the manufacturing, or group with the others as an aggregate concrete during the construction. As consequence, the aggregation level of communicating concrete in McBIM project is the intelligent container.

### 1.5 Proposed Holonic Approach and related research issues

The digital part example for the management of real materials is shown in Figure 15. Real communicating nodes (CNs) in real concrete are represented by virtual agents in the digital concrete. There are some benefits: firstly, these agents can store monitoring and product-related information to overcome the storage limitation of nodes; Secondly, this digital concrete allows to visualize energy evolution of the embedded WSN based on the received information which can be helpful for the maintaining or the exploitation. Moreover, agents can be grouped into a concrete agent which has the global view of the embedded WSN. This concrete agent can evaluate available collection strategies and then provide energy efficient solutions. By this way, this digital concrete overcomes the limit of processing power at nodes and increases intelligence level of communicating concrete.

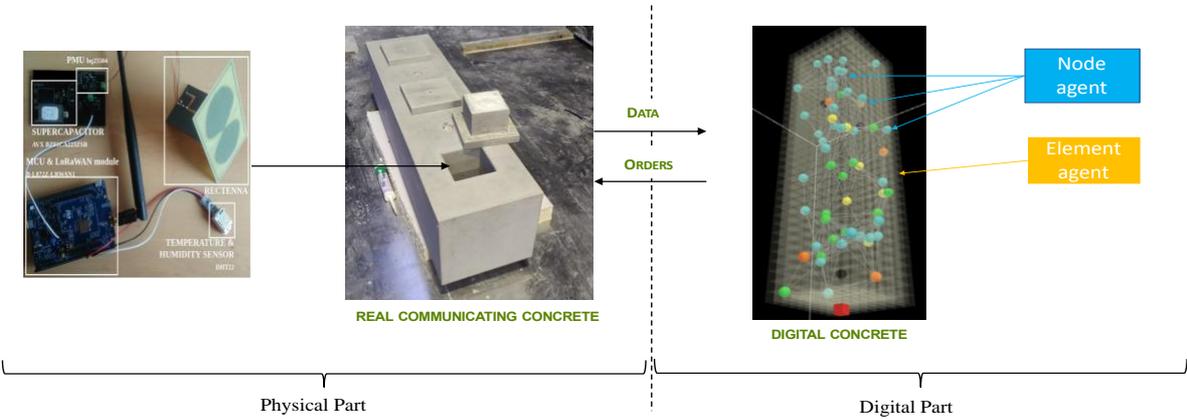


Figure 15 Digital part design for communicating concrete

To construct the digital concrete and achieve an energy efficient management, the proposed holonic control architecture is illustrated in Figure 16. The physical part monitors physical parameters and transmits to its dedicated agent. Subsequently, those data are processed by its agent located in digital part. The agent can then provide optimal data collection solutions based on received network information. The auto-organization of WSN makes the aggregation of the McBIM elements A and B possible, they can then be viewed as a whole element C. Same for the digital part, each concrete agent can work alone or can be assembled with another concrete agent to construct a higher-level agent A. Therefore, the holon C can be seen as the composition of the holon A and the holon B.

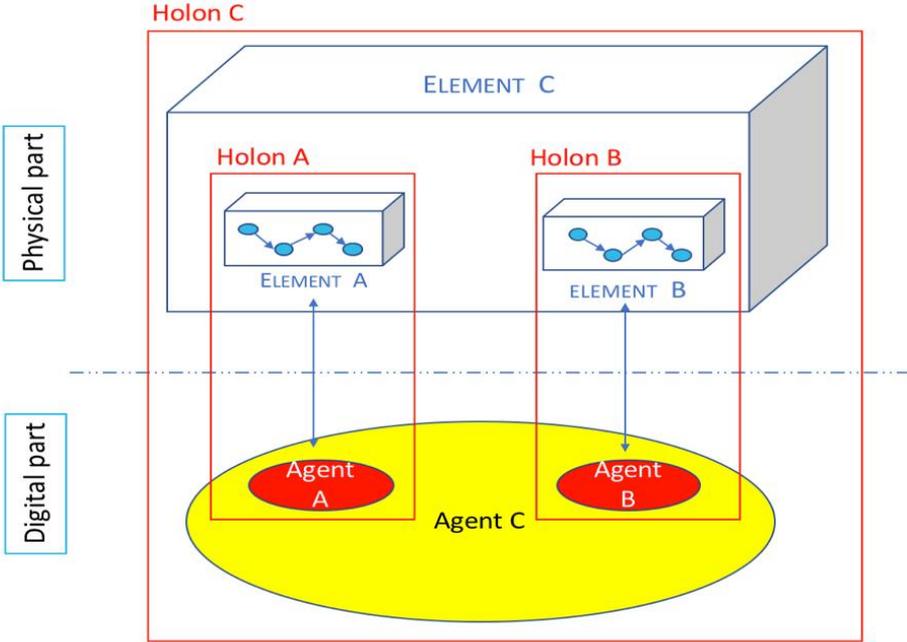


Figure 16 The holonic approach for communicating concrete

In this chapter, the communicating material paradigm was presented. The McBIM project proposes to apply the CM concept to the construction industry and to develop communicating concrete. The concept of communicating concrete has been defined and a holonic architecture is introduced for the management of its physical part and its informational part. Intelligence has been discussed and compared with previous CM applications by applying the classification proposed by Meyer (Meyer et al., 2009). To achieve the design of the concrete CM, research issues are splitted in two parts: the physical part (energy assessment and saving in WSN) and the digital part (recursive architecture for the management of agents).

For the physical part, communicating concrete should work as long as possible (from manufacturing to exploitation). To overcome the battery limit in a resource constrained environment, using efficient data collection strategies can reduce energy consumption. Besides, knowing network energy level can be helpful to change data collection strategy to extend network lifetime. Therefore, the first research issue is: **how to maximize lifetime of data collection within the embedded WSN and predict remaining energy of network with a simple analytical energy estimation model?**

For the digital part, due to the limitation of processing power and storage capability at nodes, the consistency of the information throughout the lifecycle is a problem. Besides, WSN's decentralized self-organizing solutions can only provide local optimizations. Building a digital model allows storing and presenting all information about the physical part. This approach could also provide optimal behavior decisions (with a centralized solution) to extend its lifetime. In addition, this digital model can be developed to digital twin in the future work. Realizing the digital concrete leads to the research issues: **how to manage monitoring data and energy information at agents?** For the composition of communicating concretes, the embedded networks can be grouped into one network. As consequence, the dedicated agents in digital part should also be grouped into aggregated agents. Furthermore, as required by the holonic paradigm, this agent architecture must support recursion. Therefore, **how to organize the agents to achieve a recursive architecture and how to realize the composition or decomposition?** is another research issue.

In the following chapter, literature solutions around these research issues will be reviewed. For the physical part, the state of the art around energy saving and assessment in WSN is presented in the *section 2.1*. The proposed analytical energy estimation model is detailed in *Chapter 3* and its validation is in *section 5.1*. For the digital part, recursive holonic and MAS architectures are compared in *section 2.2*. The proposed recursive MAS architecture is detailed in *Chapter 4* and its software application and validation in *section 5.2*.



# Chapter 2

## The state of the art on physical/digital part

### Contents

<a href="#">2.1 Physical part: Energy saving and estimation in WSN</a> .....	28
<a href="#">2.1.1 In-network data reduction</a> .....	30
<a href="#">2.1.2 Data aggregation energy-efficient routing protocols</a> .....	32
<a href="#">2.1.3 Time synchronization</a> .....	42
<a href="#">2.1.4 Energy consumption models for WSN</a> .....	45
<a href="#">2.1.5 Synthesis</a> .....	47
<a href="#">2.2 Digital part: energy management for communicating concrete</a> .....	49
<a href="#">2.2.1 Holonic Control Architecture (HCA)</a> .....	49
<a href="#">2.2.2 Multi-Agent System (MAS)</a> .....	51
<a href="#">2.2.3 Recursive holarchy or MAS</a> .....	53
<a href="#">2.3 Synthesis</a> .....	59

To address the mentioned research issues: energy saving and network organization in WSN; energy management of physical part in MAS, this chapter presents the state of the art on the physical part in *section 2.1* and on the digital part in *section 2.2*.

To extend the network lifetime and build an analytical energy model for the physical part, energy saving solutions are reviewed. We focus especially on three aspects: data reduction, energy-efficient routing and time synchronization in *section 2.1.1*, *section 2.1.2* and *section 2.1.3* respectively. Besides, energy consumption models are reviewed in *section 2.1.4* to build an analytic energy model for remaining energy prediction. As a conclusion, a synthesis for the physical part literature is given in *section 2.1.5*.

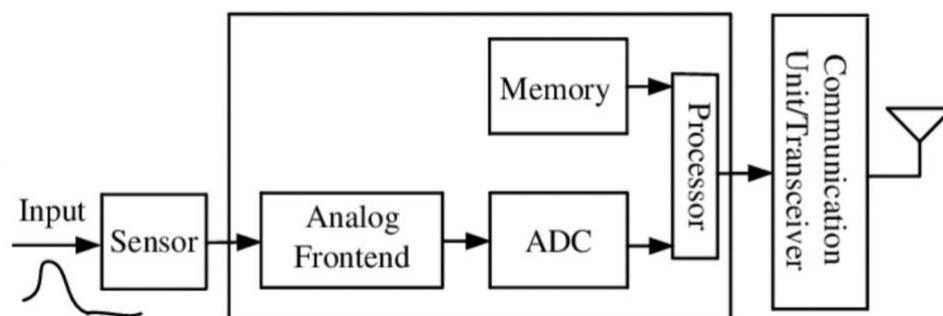
On the digital part literature, our objective is to build a recursive architecture for the energy management of communicating concrete. To achieve this goal, representative holonic control architectures are introduced in *section 2.2.1*, multi-agent system (MAS) paradigms and main adapted works are given in *section 2.2.2*. Due to the CM composition/decomposition property, the supporting MAS needs to support of recursive functions. Therefore, recursive holarchy or

multi-agent architectures are presented in *section 2.2.3*. Finally, a general conclusion on the state of the art is presented in *section 2.3*.

## 2.1 Physical part: Energy saving and estimation in WSN

To better understand energy consumption of sensor nodes in WSN, it is necessary to detail the energy consumption of their components. A sensor node has three main components (see Figure 17):

- *Sensor*: it measures physical phenomena of surrounding environment, for example, temperature, humidity, corrosion, etc. Analogical sensed data are then converted into digital data and transferred to the processor.
- *Central Processing Unit (CPU)*: it is the core part of the sensor node which manages the functionalities of all components (sensors, communication module and so on). Thanks to its memory and information processing capabilities, monitoring data can be stored locally or transmitted to other nodes via the communication unit.
- *Communication unit*: it has two main functions, receiving and transmitting data from/to other nodes.



*Figure 17 Architecture of sensor node*

Indeed, the energy consumed by a node during data collection can be decomposed in three parts: monitoring physical phenomena by sensors, data processing by CPU, data transmission by the communication module. Among them, communication unit generally consumes more energy than the other components.

As described by (Rault, 2017), there are five energy efficient ways to extend WSN lifetime as shown in Figure 18: Radio optimization, Data reduction, Sleep/wakeup schemes, Energy-efficient routing and Battery repletion.

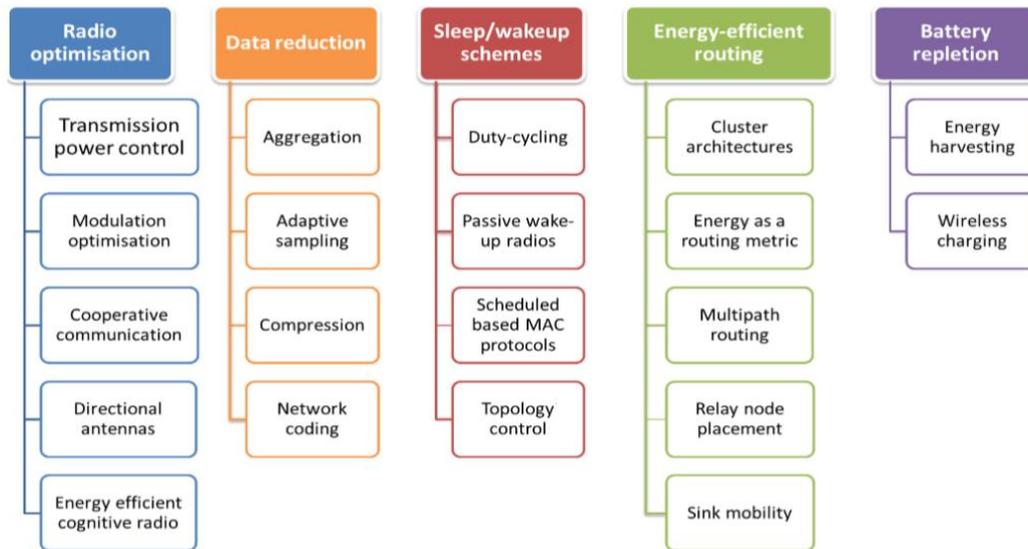


Figure 18 Classification of energy efficient mechanisms for WSN (Rault, 2017)

Over last decades, some research works focused on reducing the energy cost of radio, using adaptive power control (Teng et al., 2018), or energy efficient radio (Naeem et al., 2013). Meanwhile, lot of research focus on improving communication efficiency at nodes (Alfayez et al., 2015). To do so, some approaches as in-network processing techniques try to minimize the number of generated messages and data reduction functions are implemented at intermediate nodes to reduce the size of transmitted messages (Fasolo et al., 2007). At the same time, some approaches reduce active duration at nodes by optimizing communication schedule (Pal and Chatterjee, 2014) (Afroz and Braun, 2020). Besides, some approaches based on WSN topology help to reduce gathering path from source nodes to the sink node (Ogundile and Alfa, 2017). In addition, some works focus on power supply to overcome the limitation of resource, such as energy harvesting techniques proposed in (Kaur et al., 2019) and (Loubet et al., 2019).

The methods for radio optimization and battery repletion focus on reducing the energy needs of hardware (the design of efficient communication module and the processor). However, the used hardware depends on its application. In our study, we would like to propose a generic energy saving solution which can be applied for all constrained applications. As consequence, instead of studying energy saving for a special kind of hardware, **we focus on optimizing the energy cost of communication by applying in-network data processing techniques.**

(Kalpakis and Tang, 2009) states that Maximizing Lifetime using Data Aggregation (MLDA) requires to take simultaneously into account the data aggregation technique, routing protocol

and time synchronization as shown in Figure 19. As a result, the main works proposed in the three aspects are described hereafter. As said earlier, energy optimization could also require monitoring the energy consumption. Energy models related to node and data collection are also reviewed in the following.

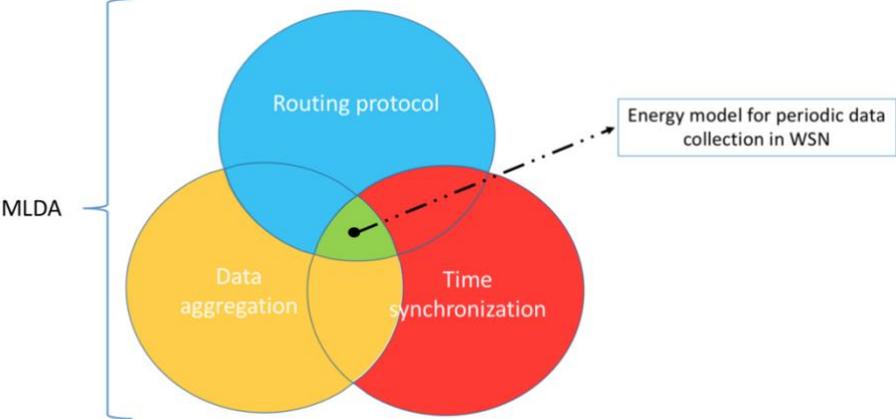


Figure 19 MLDA problem in WSN

2.1.1 In-network data reduction

In order to reduce energy cost for communication at nodes, in-network data reduction functions are implemented at intermediate nodes to process raw data. These functions aim to reduce the size of data to be delivered. Thanks to data aggregation functions, only reduced messages are transferred to next nodes until the sink (Yao and Gehrke, 2002). An example is shown in Figure 20. Instead of transferring all received data, original messages are merged or reduced to an aggregated message at intermediate node to reduce communication cost.

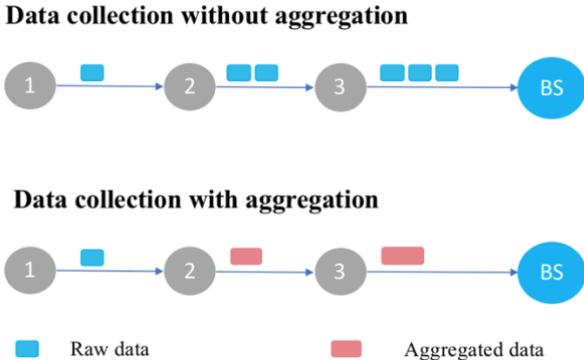


Figure 20 Data aggregation example

Aggregation functions can be classified into two main approaches depending on the processing way: lossy data aggregation or lossless data aggregation. Basic *lossy aggregation* functions are simple operations on data like “Average, Max or Min” etc. These operations reduce

transmission cost, but this reduction is obtained by losing data accuracy (Cui, 2017). For example, a network monitors and reports average temperature of its nodes and only the average temperature is sent. At the end, the sink node only knows the average temperature of the network. The original information can not be recovered.

In contrast, data collection with *lossless aggregation* compresses or merges data while preserving all original data. In general, the packet format for communication contains two parts (header and data) as shown in Figure 21. The header part contains numbering and addressing, for example, node ID, request ID, addresses and etc... The sensor reading part contains sensor data. In the case of a periodic request, most of the information in the packet header is the same with all reads, and therefore can be shared for data collection with lossless aggregation method (data fusion) as in (Jun-Zhao and Jiehan, 2008). However, the energy cost for transmission is higher because there are more data to be transferred.

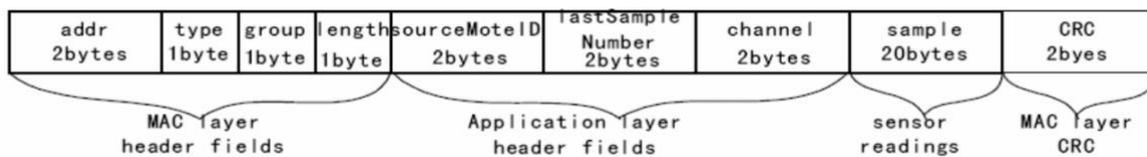


Figure 21 The packet formats in application OscilloscopeRF (Jun-Zhao and Jiehan, 2008)

Data compression is another reduction technique, it reduces the size of data by using compression algorithms (Sheltami et al., 2016). Although energy cost for data transmission is reduced with compression method, accessing memory during the compression process increases runtime at nodes. Therefore, the selection of appropriate compression algorithm depends on the accuracy requirement.

Besides, some predictive models are used for practical monitoring application (J. Lue et al., 2010) (Wei et al., 2011). The idea is to use received monitoring data to predict new data when the required accuracy of predictive is sufficient. For example, as described in (Cui, 2017), sensor nodes do not need to send all the raw data if the sink can recover the high-fidelity data from the predictive models.

To summarize, all the mentioned reduction methods aim to reduce the size of transmitted data to extend network lifetime. Although communication activities are decreased with data reduction techniques, energy cost for data processing (fusion or compression) could not be

ignored. The trade-off between processing power requirement and energy efficiency is shown in Figure 22. Compared with data aggregation and data compression, data prediction is an out network processing method which has higher processing power requirements and is generally performed in cloud. Although prediction or compression methods could greatly reduce the size of transmitted data and so the radio activity duration, the time for message accessing and processing could not be ignored. This could also increase energy cost and shorten the network lifetime.

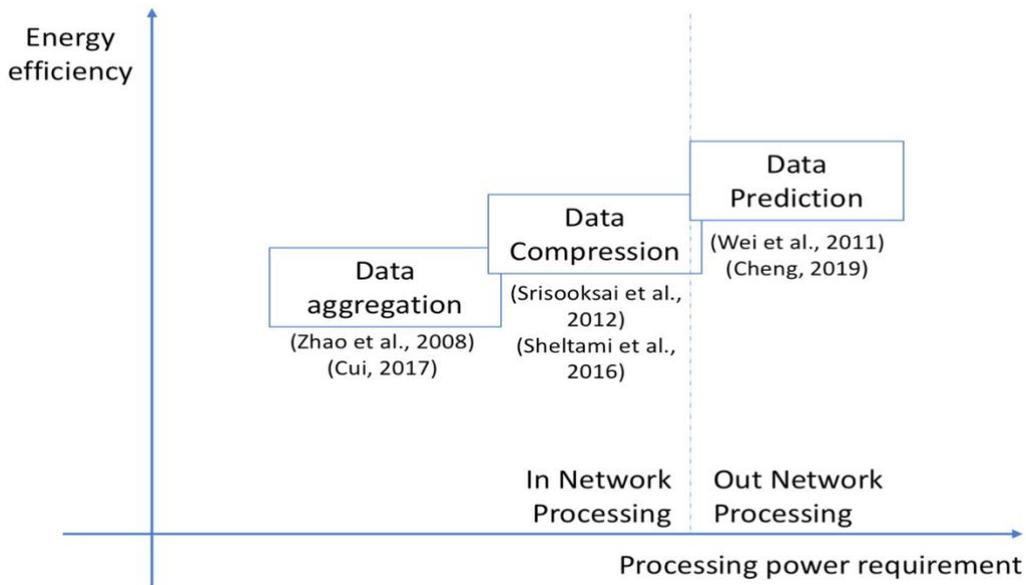


Figure 22 Tradeoff of data reduction functions

In a resource constrained environment, the energy supply of node is limited. As consequence, the trade-off between energy efficiency and processing power requirement should be considered for selecting suitable data reduction functions (Jun-Zhao and Jiehan, 2008) (Sheltami et al., 2016) (Cui, 2017).

### 2.1.2 Data aggregation energy-efficient routing protocols

A short gathering path helps to reduce the transmission cost from source nodes to BS. As a result, many routing protocols for data collection have been proposed since the last two decades. Concerning the data gathering structure, routing schemes can be classified in two categories: flat routing and hierarchical routing (Khedre et al., 2021). In flat-based routing, messages are flooded between sensors nodes until reaching the sink. In the second one, message transmission follows a hierarchical structure, which allows to define guided paths to the BS and limit the

energy consumption of the network. There are three kinds of structures used for hierarchical topology as shown in Figure 23: *cluster-based*, *chain-based* and *tree-based* routing structures.

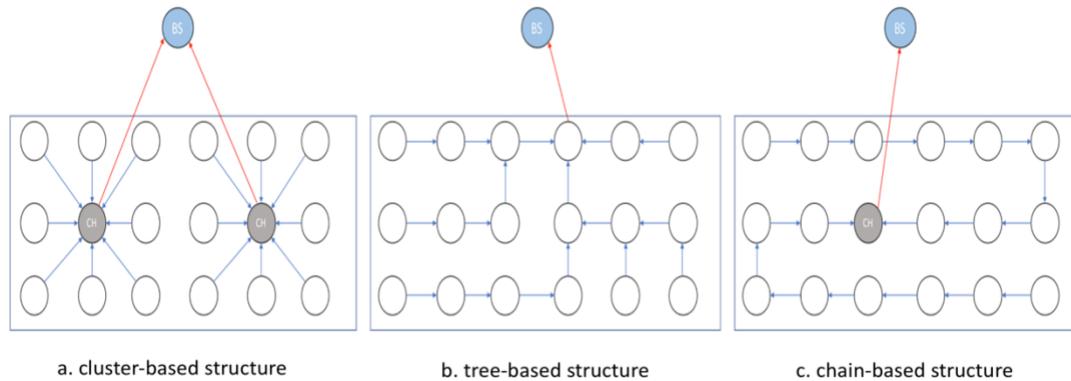


Figure 23 Hierarchical data collection structures in WSN

In a chain-based WSN, nodes transmit information to BS through a linear structure (Lindsey et al., 2002) (Rani et al., 2015). In a cluster-based architecture, few nodes are elected as cluster head (CH) by a selection algorithm for each cluster. Cluster heads aggregate information from other cluster members and transmit aggregated data to BS (Heinzelman et al., 2002) (Braman and Umaphathi, 2014). In a tree-based structure, nodes transmit information along a tree structure as in (Tan and Körpeoğlu, 2005) (Han et al., 2014). Some representative routing protocols from the three structures, adapted to data aggregation, are presented and discussed hereafter.

### Cluster-based routing protocols

LEACH (Low Energy Adaptive Clustering Hierarchy) is the first proposed cluster-based protocol (Heinzelman et al., 2000). As illustrated in Figure 24, it is a distributed and self-organized clustering protocol. A random mechanism is used to rotate the CHs in order to balance the energy consumption.

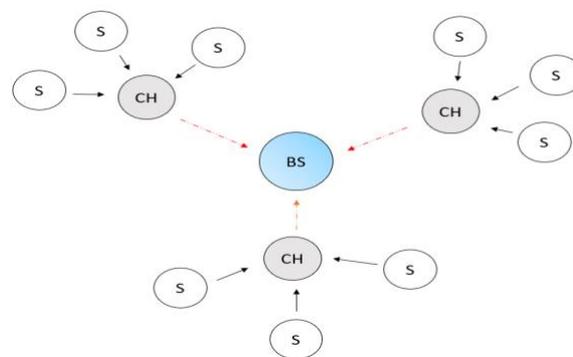


Figure 24 LEACH routing protocol

LEACH works with two phases per round: *setup phase* and *steady state phase*. During the setup phase, all nodes organize themselves into clusters. A distributed probabilistic approach is used to elect cluster leader. The threshold  $P_i(t)$  is defined as in (1):

$$P_i(t) = \frac{k}{N - k * (r \bmod (N/k))} * c_i(t) \quad (1)$$

Where  $k$  is a predetermined cluster number,  $N$  is the total number of nodes,  $r$  is the current round number,  $i$  is the node's identification (ID) and  $c_i(t)$  is a function which indicates if the node has already been a CH in a previous round. For each round, sensor nodes randomly pick a number from 0 to 1, and then compare the picked value with the threshold. A node acts as leader if its picked number is bigger than the threshold. This probabilistic approach balances energy consumptions within clusters. When a CH is elected, it broadcasts a message to announce that it is the cluster leader. Surrounding nodes then decide the cluster to join, the decisions depending on the signal strength of received messages. The second phase of the process is data collection. All non-headers transmit data to its CH. Once cluster leaders receive all data from their cluster members, they can then transmit data directly to the sink. This algorithm improves the lifetime of network, but the single-hop routing is inefficient for a large network. The leaders who are far from the sink consume a lot of energy. Furthermore, aggregation function could be implemented at cluster-heads to reduce transmission cost. Based on LEACH, the authors also proposed a centralized method LEACH-C where the clustering of nodes is performed at the sink to improve the performance (Lindsey et al., 2002).

Unlike the random election of leader in LEACH, Hybrid Energy-Efficient Distributed clustering (HEED) takes residual energy and communication cost into account (Younis and Fahmy, 2004). An Average Minimum Reachability Power (AMRP) is defined for communication cost estimation. It is the average of minimum power levels for the connection from non-leader nodes to its CH. In each round of data collect, every node computes its probability  $P_{CH}$  which is defined as in (2). A node becomes the leader if its  $P_{CH}$  reaches 1.

$$P_{CH} = C \times \frac{E_{residual}}{E_{max}} \quad (2)$$

Where  $C$  is the defined percentage of cluster heads,  $E_{residual}$  is the current residual energy, and  $E_{max}$  is the initial energy. Once the leaders are elected, the surrounding nodes can then choose a CH with the lowest AMRP cost. Compared with LEACH, this residual energy-based election mechanism avoids excessive energy consumption of some nodes and thus extends network lifetime. An improved algorithm, DWEHC (Distributed Weight-based Energy-efficient

Hierarchical Clustering protocol) is proposed by (Gupta and Younis, 2004). Like HEED, the residual energy is accounted in DWEHC. Within cluster, the no-leader nodes use the knowledge of distance to decide to be a *single-level* (transmit data directly to CH) or *multi-level* member (transmit data via another member to CH). This location awareness balances the cluster size and improves the energy efficiency.

Over the last two decades, some LEACH-based extensions have been proposed to improve energy efficiency (Singh et al., 2017). Such as TL-LEACH in (Loscrì et al., 2006), LEACH-CC in (Yassein et al., 2009), W-LEACH in (Abdulsalam and Kamel, 2010) and so on. The main differences between cluster-based protocols can be summarized in three aspects: the selection mechanism for the CHs, the inter-communication between CHs and the sink, the intra-communications between the cluster members and their CH.

### **Tree-based routing protocols**

Another widely used approach in WSN is the tree-based topology, where sensor nodes transmit data following a tree structure network via relay nodes to the sink. Processing operations such as data aggregation can be performed at intermediate nodes to reduce data transmission and save energy.

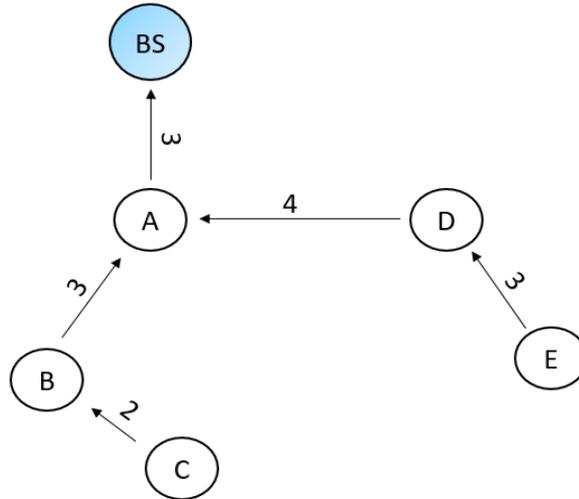
The Tiny AGgregation (TAG) approach is a data-centric tree-based protocol (Madden et al., 2002). A declarative interface is used for data collection and aggregation. With a given query, relevant information is sent to the sink periodically. This algorithm has two phases: construction phase and collection phase. In the first phase, the sink broadcasts a message which contains the distance to sink. When a node (not already in the tree) receives a message, it sets the sending node as its parent and then broadcasts again the same message with its own ID and distance. After the tree is constructed, a semantic query which specifies required data (such as temperature or humidity) is sent from the sink to all nodes. Relevant nodes send data to its parent then go to sleep. A parent node fuses all received data with its own and transmits the aggregated message. This in-network aggregation processing algorithm reduces transmission at intermediate nodes and improves network lifetime. However, it is not robust to link failure and re-organization is highly energy consuming.

An Energy-Aware Distributed Aggregation Tree (EADAT) is proposed in (Ding et al., 2004). This approach combines residual energy of nodes and their number of hops (or distance) to the sink to construct the tree. A control message which contains five fields (*ID*, *parent*, *power*, *status*, *hopCnt*) is defined for construction. Where *ID* and *parent* record the node's identification, and its parent node respectively; *power* represents its residual energy; *status* indicates the state of the sending node in the network (undefined state, leaf node, non-leaf node, or danger state); *hopCnt* is the number of hops to reach the sink. At the beginning of the construction process, the sink node broadcasts an initial control message. A node  $v$  sets up its timer after receiving the first message. Based on the received message, the node  $v$  chooses a neighbor as its parent which has higher residual power and shortest path to the sink. It then broadcasts a message with its own *hopCnt*. All nodes will broadcast their message and find their corresponding status in the network until the construction of tree is completed. Compared with TAG, residual energy is taken into account to increase robustness and to reduce maintaining cost. However, due to the lack of a global view of whole network, it is hard to ensure that gathering paths in this distributed construction tree is optimal.

A Power Efficient Data gathering and Aggregation Protocol (PEDAP) was proposed by Tan in (Tan and Körpeoğlu, 2005). It is a near optimal minimum spanning tree-based (MST) protocol. In this approach, the computation of the routing tree is completed in a centralized manner at the BS. To construct the MST, the edge  $e_{ij}$  between nodes  $i$  and node  $j$  is computed by the equation (3). The latter presents communication cost between these two nodes.

$$C_{ij} = 2 \times E_{elec} \times k + E_{amp} \times k \times d_{ij}^2 \quad (3)$$

Where  $E_{elec}$  is energy consumption of transmission or reception,  $E_{amp}$  is consumption of transmit amplification,  $k$  is the size of packet and  $d$  is the distance. The construction begins with the root (the sink node). An MST example is shown in Figure 25. For each iteration, the root adds the vertex which has the minimum weighted edge into the tree. At the end, a network with a minimal weight route is constructed. However, the runtime complexity of network construction can not be ignored in this approach, which may increase maintaining cost, especially for a large network, it is equal to  $O(n^2)$  where  $n$  is the size of the network. The larger the network, the longer its network construction time. In order to improve the robustness of this structure, a power aware version PEDAP-PA which takes residual energy of node into account has been proposed. In PEDAP-PA, the weight is replaced by  $EC_{ij}(k) = C_{ij}(k)/e_i$  where  $e_i$  is the normalized residual energy of node  $i$ .



*Figure 25 Minimum spanning tree-based routing protocol*

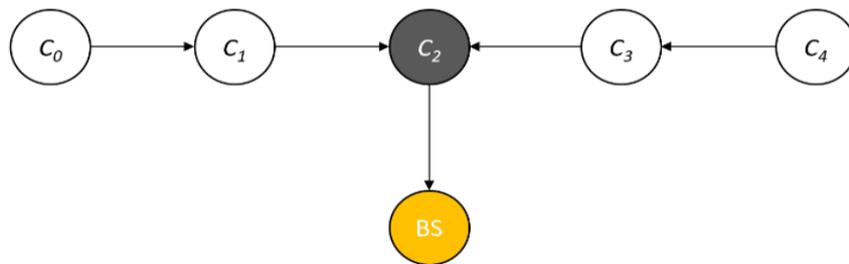
These mentioned tree-based protocols aim to construct an efficient gathering tree based on different aspects: distance, residual energy, the number of hops to sink and the total gathering path. There are many other solutions that have been proposed for tree construction in a distributed or a centralized manner, like the use of Ant Colony Optimization in (Lu et al., 2015), the residual energy considered tree in (Virmani et al., 2013), the shortest path tree for dynamic sink application in (Hawbani et al., 2018).

Compared with cluster-based, tree-based protocols aim to optimize energy consumption over whole network and network construction is always completed at sink node. However, maintaining energy cost is higher than in cluster-based structures. When an intermediate node runs out of energy, tree-based approach may need to completely reconstruct the network where only one part of the network has to be reconstructed with a cluster-based approach. Apart from these two kinds of structure, chain-based structure has also been widely applied. Some examples are given in following section.

### **Chain-based routing protocols**

In a chain-based routing protocol, data are transmitted from the furthest node to the BS through a line structure. Data aggregation can be performed at intermediate nodes to reduce communication cost and extend network lifetime.

The first historical chain-based routing protocol using data aggregation is PEGASIS (Power-Efficient Gathering in Sensor Information Systems) (Lindsey et al., 2002). In this protocol, the greedy algorithm is used for chain construction. It selects closest neighbor as next hop. During the data collection phase, each node takes turn to act as the leader with the probability  $i \bmod n$  where  $i$  is the node number and  $n$  is the number of nodes. Data collection begins with one of the end nodes ( $C_0$  or  $C_4$ ) as shown in Figure 26 and data aggregation is performed at all intermediate nodes. Once the chain leader receives all messages from a side, the collection process is then repeated for the other side. At the end, the chain leader fuses all received data with its own and transmits aggregated message to BS.



*Figure 26 PEGASIS chain-based protocol*

The random leader selection mechanism used in PEGASIS balances energy consumption at nodes. Besides, it uses in-network data aggregation to reduce the amount of transmitted data. However, a long chain structure may have a huge impact on the latency and on the delivery time.

CRBCC (Chain-Cluster Based Mixed routing) is a two-layer hierarchical protocol proposed by (Zheng and Hu, 2010). The authors suppose that each node is assigned  $(x, y)$  coordinates. During the construction, the network is divided into balanced clusters by using  $y$  coordinates. In each cluster, the Simulated Annealing (SA) algorithm is performed to construct the lowest energy consumption chain. The initial objective of SA algorithm is to compute the shortest way with the lowest energy consumption. The chain leaders in the low-level are elected in the same  $x$  coordinates order as shown in Figure 27. The SA algorithm is then performed again for the construction of leader-leader chain. At the end of the construction process, a top leader is randomly elected to communicate with the BS. In the phase of data collection, data fusion is performed in both low-level chain and leader-leader chain. Once data collection is completed, the top leader then fuses all data and transmits it directly to BS. This protocol combines the

advantages of chain-based and cluster-based protocols to overcome the poor latency and high reconstruction cost in a long chain structure.

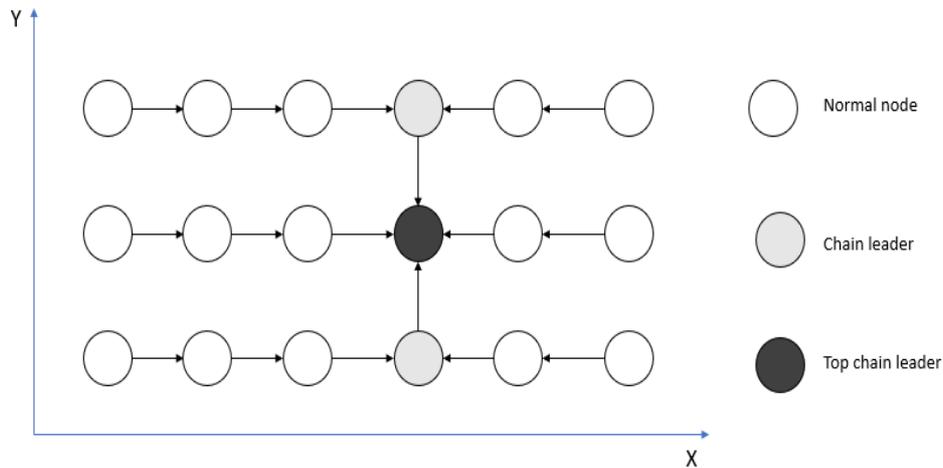


Figure 27 CRBCC protocol

To avoid high maintaining cost of long chain structures, a sub-network chain-based routing protocol (BCBRP for Balanced Chain-Based Routing Protocol) has been proposed in (Ahn *et al.*, 2011). Compared with the previous algorithms, chain construction in BCBRP is more complex. It has four phases: dividing network, selecting bridge, chain construction with MST, and selecting leader. Although data gathering in BCBRP is also along a single chain structure, chain reconstruction takes place only in the sub-network which contains a dead node. Therefore, energy maintaining cost is reduced, and thus network lifetime is extended. However, the chain construction is complex, and the drawbacks of the long-chain structure in terms of gathering delay are not solved.

In order to improve energy efficiency and extend network lifetime, many other chain-based protocols have been proposed. According to (Khedr *et al.*, 2021), recent chain-based solutions can be divided into two categories based on the intra-cluster and inter-chain communication: first, multi-hop/single-hop; second, multi-hop/multi-hop. As illustrated in Figure 28, first category protocols are in blue color, such as the AI-PEGASIS (Ndawi, 2015), DLRP (Marhoon *et al.*, 2020); and the second category in red, like the EPEGASIS (Mahdi *et al.*, 2015) or PEG-ACO in (Ramluckun and Bassoo, 2018).

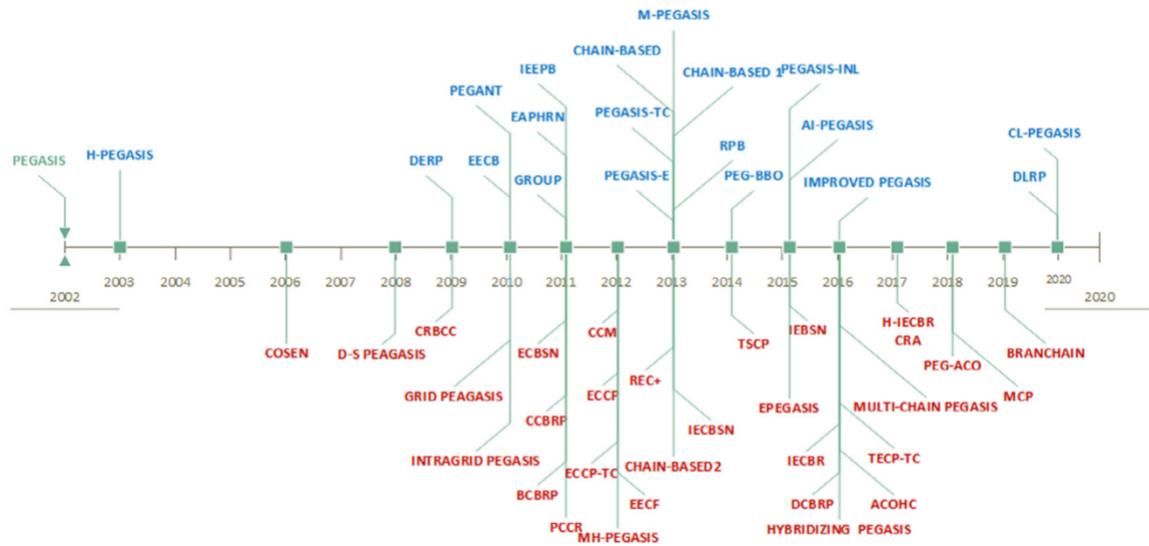


Figure 28 Timeline for chain based routing protocols (Khedr et al., 2021)

Energy saving for data gathering in WSN is still a hot research issue. In this thesis, we can not review all solutions because new protocols are proposed every week. Instead of, we would like to find out which kind of structure is suitable for communicating concrete application and for building analytical energy consumption models. Since the behavior of communicating concrete and applicative needs can be different throughout its whole lifetime, a unique routing protocol should not be suitable for all CM lifecycle phases.

During manufacturing, the communicating concrete is mainly used to monitor its status, each element may work independently. As the network of CNs within a single concrete element is not large, a simple and efficient data aggregation protocol will be preferred. For construction phase, the problems are not only the minimization of energy within the concrete, but also the adaptation of network when new concrete elements arrive. Thus, a data aggregation technique favoring dynamic and fast network reconfiguration should be privileged. After the building construction, a large network will be completed and used for the exploitation phase. Consequently, a robust topology with a low maintaining cost protocol should be more appropriate.

To sum up, the required performance metrics for data aggregation routing protocols are described as follows: energy efficiency which represents the algorithm efficiency for data collection. It is computed by the amount of consumed energy by the number of received messages; Maintaining cost shows its energy needed for the reconstruction of the architecture;

Resilience to link failure indicates its robustness facing the loss of link when a node dies; Scalability stands for its capability to scale the scenario. The different types of structures presented above are compared with the mentioned performance metrics in *Table I*. The plus and minus are used to represent the strengths and weaknesses of protocols.

Table I. Summary of data aggregation protocols

Protocol name	Structure-based	Energy efficiency	Maintaining Cost	Resilience to link failure	Scalability
LEACH	Cluster	--	+	+	--
HEED	Cluster	+	-	+	+
DWEHC	Cluster	++	-	++	+
EADAT	Tree	-	--	-	+
PEDAP	Tree	++	+	-	+
TAG	Tree	+	++	-	+
PEGASIS	Chain	-	--	-	--
CRBCC	Chain	+	+	+	-
BCBRP	Chain	-	--	++	--

All mentioned protocols are compared to find which kind of structure is suitable for each lifecycle phase. As mentioned above, the communicating concrete mainly monitors its status during the manufacturing. Therefore, a simple and efficient data aggregation protocol should be preferred. We can see from the table that PEDAP for tree structure and CRBCC for chain structure have high energy efficiency and low maintaining cost. Meanwhile, the construction of chain and tree structure is easier and less energy intensive than for cluster, therefore, a chain or tree structure may be preferred for the manufacturing.

During building construction, concretes should "cooperate" and the network structure could be modify. Thus, a data aggregation technique favoring dynamic and low maintaining cost network should be privileged. Compared with chain-based structures, cluster-based approaches (like HEED) and tree-based approaches (like TAG) provide a good scalability. However, the high maintaining costs in cluster-based structure are not suitable for this phase. In contrast, PEDAP and TAG (tree-based) have a low maintaining cost with a good scalability and high energy efficiency and thus may be a good choice for the construction phase.

After the construction, a large network will be completed and used for building's exploitation. An efficient, robust and low maintaining cost protocol should be adapted. CRBCC may have good result, however, the delay delivery drawback of a long chain structure cannot be neglected. Cluster-based structure have better resilience to link failure than tree-based. An energy efficient cluster-based protocol such as DWEHC may lead to good results for periodical monitoring. Tree-based routing protocol like TAG should be considered for data-centric applications.

These three types of structures have their own advantages and drawbacks according to the CM lifecycle phase. We will use these different types of structures in communicating concrete lifecycle. A lot of these protocols are based on local decisions taken by nodes. Indeed, each node independently reacts, by considering its own states and its neighborhood, to take routing decisions. This leads to local optimal solutions but could result in suboptimal global ones. These decisions may be more optimal if taken in the digital part, in a centralized manner, that may help to reach a global optimum. Meanwhile, energy cost for inactive state is often ignored in these routing protocols which should be also taken into account for energy estimation. To ensure nodes can wake up at the right moment, some time-synchronization solutions are thus presented as follows.

### 2.1.3 Time synchronization

As said previously, using sleep mode on radio module allows to save energy at nodes. However, to ensure the communication with sleep mode, the time synchronization problem should be considered. Therefore, some representative time synchronization protocols for WSN are presented and discussed hereafter. MAC protocols can be classified in three groups according to synchronization mechanism: *sender-based synchronization model*, *sender-receiver interaction model* and *receiver-receiver interaction model*. In this section, representative protocols for each model are presented and discussed.

DMTS (Delay Measurement Time Synchronization) is one of the **Sender-based synchronization models** (Ping, 2003). For a one-hop network, fewer messages are transmitted. The synchronization message that contains the global time  $t$  is broadcasted by the BS. Each child node records the message arrival time  $t_1$  and the time before adjusting its clock  $t_2$ . Besides, the delay for transmission  $t_e$  can be estimated by the transmission rate of radio module, such as

20 kbps with Mica. Therefore, the delay  $t_{delay}$  is presented as in (4). Child nodes then adjust their local times to the global time with measured path delay.

$$t_{delay} = t_e + (t_2 - t_1) \quad (4)$$

This one-hop method can achieve high energy efficiency. However, it is hard to ensure that each node can directly communicate with the BS. Therefore, the authors also discussed the multi-hop version. It uses source levels to identify the distance from the master to another node. Source level of the leader is zero. The synchronization process works as follows: first, a leader selection algorithm is used to select a time master. The leader then broadcasts its time to other nodes. When receiving a time signal, a node checks the level of the time source  $n$ . If it is from a source of lower level than itself, it accepts the time and set its source level to  $n + 1$ . Otherwise, it silently discards the signal. The root node periodically broadcasts its time. All nodes are synchronized by the multi-hop method.

Reference Broadcast Synchronization (RBS) was proposed by Elson et al. (Elson et al., 2002). Unlike the previous one, this method is based on a **receiver-receiver interaction**. In this protocol, the reference node (call sender in Figure 29) broadcasts a reference packet to others. When node A receives the packet, it records its local time at  $T_{21}$ . Meanwhile, node B also records its local receiving time  $T_{22}$ . The nodes can then synchronize by modifying their own time according to the time difference  $m$  as in (5).

$$m = T_{21} - T_{22} \quad (5)$$

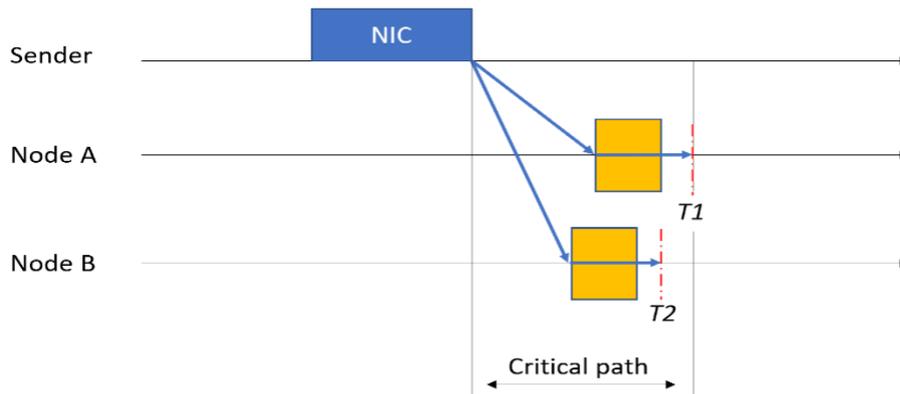


Figure 29 RBS method

Multi-reference packets are also used to improve the precision of synchronization between two nodes. The transmitter broadcasts  $m$  reference packets, any receiver node  $i$  can synchronize itself with another receiver  $j$  by exchanging all its receiver times  $T_{i,k}$  and  $T_{j,k}$  where  $k \in m$ . The

offset formula is shown in (6). Besides, extension version is also discussed for multi-hop network.

$$Offset[i, j] = \frac{1}{m} \sum_{k=1}^m (T_{j,k} - T_{i,k}) \quad (6)$$

In traditional synchronization protocols, the critical path contains the send time, access time, propagation time, and receive time. However, only propagation time, received time and its message arrival time are exchanged with others to realize the synchronization. By this way, it reduces the send time and access time to improve the accuracy.

Timing-Sync Protocol for Sensor Networks (TPSN) is a **sender-receiver interaction** method proposed by Ganeriwal (Ganeriwal et al., 2003). This protocol is inspired by the NTP (Network Time Protocol) which has been widely used in Internet. A two-way time exchange method is used for synchronization as shown in Figure 30. Node A first sends a synchronization message that contains its time  $T_0$ . Node B records the received time  $T_1$  that is the sum of  $T_0$ ,  $\alpha$  and  $d$ . Here,  $\alpha$  represents the clock drift between the two nodes,  $d$  is the propagation delay. B then sends back an acknowledgement message that contains the three times  $T_0$ ,  $T_1$  and the sending time of message  $T_2$ .

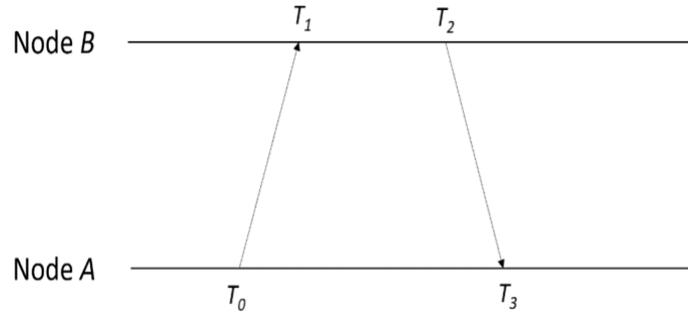


Figure 30 Synchronization between two nodes in TPSN

Node A records the received time  $T_3$ , it can then calculate the clock drift and the propagation delay with the four timestamps as in (7) and (8):

$$\alpha = \frac{(T_1 - T_0) - (T_3 - T_2)}{2} \quad (7)$$

$$d = \frac{(T_1 - T_0) + (T_3 - T_2)}{2} \quad (8)$$

The authors also introduce the notion of real time for error analysis. All packet delays at sender, propagation of messages and receiver are taken into account. Compared with RBS, this method

gives better accuracy, but it consumes more energy. In the following section, energy models related to node and data collection are presented.

### 2.1.4 Energy consumption models for WSN

The distance-based dissipation First Order Radio Model (FORM) in (Heinzelman et al., 2002) has been widely used for energy consumption estimation. It is based on the distance between a transmitter and a receiver and on the size of the data to be transmitted (see Figure 31). In this model, the transmission cost  $E_{Tx}$  of a  $k$ -bit message with distance  $d$  is given by (9).

$$E_{Tx}(k, d) = \begin{cases} kE_{elec} + k \epsilon_{fs} d^2, & d < d_0 \\ kE_{elec} + k \epsilon_{mp} d^4, & d > d_0 \end{cases} \quad (9)$$

Where,

- $E_{elec}$  is the transmitting circuit energy loss,
- $\epsilon_{fs}$  and  $\epsilon_{mp}$  are the amplifier energy for free space and multipath fading respectively,
- $d_0$  is the distance threshold.

The receiver energy consumption depends only on message size  $k$ :  $E_{Rx}(k) = kE_{elec}$ . This model has been widely used to estimate radio node energy consumption (Younis and Fahmy, 2004) (Jan et al., 2013). However, some authors argue that the cost related to state transitions of the communication module should also be taken into consideration (Margi and Obraczka, 2004) (Howitt et al., 2005).

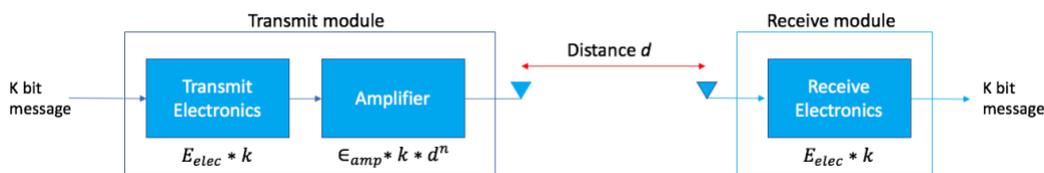
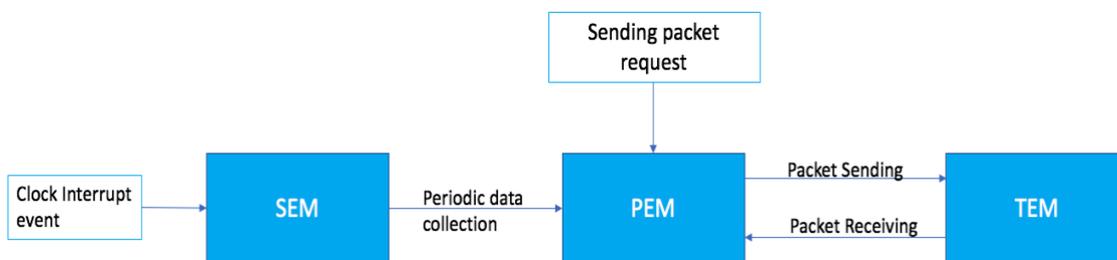


Figure 31 Radio energy dissipation model

Wang et Yang propose a hardware-based model: CSESM (Communication Subsystem Energy Consumption Model) (Wang and Yang, 2007). In this model, the main energy cost related to the main sub-modules of the communication system (receiving module, transmitting module, voltage regulator, crystal oscillator, bias generator, and frequency synthesizer) are considered. Besides transmitting  $Tx$  and receiving  $Rx$  states, three more states (power off, power down and power save) are added according to the system state machine. Although transmission modules account for a large part of energy consumption, other components should be considered. Du et

al. present an energy model that considers microcontroller and transceiver (Du et al., 2010), where all related state transitions are taken into account for the computation of energy consumption.

An event-trigger based energy model is presented by Zhou (Zhou et al., 2011), which includes processors, RF modules and sensors. The authors define four different energy models: Processors Energy Model (PEM), Transceiver Energy Model (TEM), Sensor Energy Model (SEM) and Node Energy Model (NEM). In these models, different states of processor and transceiver are considered, as well as the state transitions. With the proposed event-trigger mechanism as in Figure 32, energy consumption of nodes can then be computed. All mentioned models take one or the main components of node to estimate node and network lifetime.



*Figure 32 Event trigger mechanism of NEM*

Other approaches are based on the actual consumption of a sensor node by considering finely the different elements consuming energy in a sensor node (Srbinovska et al., 2017) (Jalsan et al., 2014). A first level of classic decoupling between CPU, sensor and radio transmitter reveals that the radio transmitter can be responsible for more than 80% of the global consumption (it is true in the majority of cases, except using very complex data reduction techniques or energy-intensive sensors like chemical sensors). These models are often used as the decision metric of the construction or organization algorithms of the WSN. They are theoretically true and may give very good approximations for wireless communications.

Some energy consumption evaluation models or lifetime estimation models have been proposed for data gathering in WSN. For example, the energy evaluation model proposed by Jan for data gathering in centralized clustering structure (Jan et al., 2013). The authors propose a cluster structure which has three types of nodes: the sink node, high-energy nodes as cluster leader and sensor nodes. They apply the FORM model to estimate the energy consumption of the three

types of nodes. The FORM model is also used in (Ren et al., 2016) to estimate the entire network lifetime. For a given network, the authors analyze traffic load to compute the number of data packets to be transferred at nodes, which is then used to estimate energy consumption and calculate network lifetime.

Although the FORM has been widely applied for network lifetime estimation, the distance is not always available in some applications and energy consumption in sleep or idle should be also considered. (Dron et al., 2014) present an emulation-based method for lifetime estimation. The authors use MSPSim/Cooja emulator to obtain energy consumption information of different devices as shown in Figure 33. This energy information is then used in a network simulator to estimate energy consumption in a special data gathering structure.

MSP430	Typical		Worst		Average	
	2.2V	3.0V	2.2V	3.0V	2.2V	3.0V
Voltage						
LPM3 ( $\mu\text{A}$ )	1.1	2.0	1.6	2.6	1.35	2.3
LPM0 ( $\mu\text{A}$ )	50	75	60	95	55	85
Active@3.9Mhz (mA)	1.287	1.95	1.56	2.343	1.4235	2.145

CC2420	Typical
Power Down ( $\mu\text{A}$ )	20.0
Idle (mA)	0.426
RX (mA)	18.8
TX@0dBm (mA)	17.4

Figure 33 MSP430 and CC420 current consumption information (Dron et al., 2014)

Some other lifetime estimation methods have been also proposed for energy consumption of node or data gathering in WSN. (Ferry et al., 2011) present a power/energy estimator for designing WSN nodes where components of node and battery are both taken into account to determine the lifetime. (Srbinovska et al., 2017) present a power-based energy model which compute the time in different states (active and sleep) to estimate energy consumption at nodes. (Vanjale et al., 2020) propose a network lifetime model which takes the residual battery capacity, total energy consumption by microcontroller and transceiver at node and discharge rate of battery into account to estimate node lifetime.

### 2.1.5 Synthesis

Using data reduction techniques at nodes allow to reduce transmission messages and extend network lifetime. Each technique has advantages and drawbacks. Although approximation or

compression methods could greatly reduce the size of transmitted data, the time for message accessing and processing can not be ignored which increases the energy cost at node. Data aggregation techniques reduce the size of transmitted data, but all data may not be reconstructed in the case of lossy aggregation. It is the trade-off between accuracy and energy efficiency. Due to the limitation of battery and processing power of node, data aggregation methods will be preferred for the application of communicating concrete. This approach should reduce transmitted data but does not significantly increase the cost of data processing.

Grouping nodes in a short gathering path reduces transmission cost from source nodes to the BS. The chain-based approach is easy to implement and very adapted for small networks, but it may cause a long delay. The strong scalability of tree-based protocol makes the auto-organization of communicating concretes possible for the construction phase. However, low resilience to the link failure, and high maintaining cost are unavoidable drawbacks. Cluster-based approach provides a robust network and low delay for a large network, it may be a good choice for the building exploitation phase.

The mentioned time synchronization protocols enable time scheduling which provides an efficient way of using radio sleep mode at nodes. In our applicative case, the network within one McBIM element is not so large. Due to the influence of material, the communication range in concrete is shorter than that in the air. As consequence, a send-receiver based method is more suitable than the two other types. Based on this approach, processes for collection in a chain structure network with or without aggregation are analyzed and models for energy consumptions are constructed in next chapter.

Distance-based energy models are widely used for the network structuration. Distance between nodes is not always suitable for WSNs, especially for distributed algorithms, because positioning is an intensive-energy task for sensor nodes. Most of time, positioning in WSNs is not possible or energy unrealistic. For the communicating concrete application, even if positions of nodes can be known because they can be manually deployed, classical propagation models (free space or multipath models) can not be applied because sensor nodes are placed in concrete (heterogeneous environment).

In conclusion, to extend embedded network lifetime, different routing protocols may be applied over the whole lifecycle of communicating concrete. Using in-network data aggregation

reduces communication consumption, and synchronization method should be considered for ensuring the communication between nodes. Both these energy saving solutions are considered for building the dedicated energy model which is detailed in *chapter 3*. To overcome the limitation of processing power and storage capability at nodes, intelligent management solutions can be added to improve further energy efficiency of the network. Representative approaches are detailed and discussed in the following section.

## 2.2 Digital part: energy management for communicating concrete

To realize the holonic architecture for the communicating concrete, this section begins with the representative traditional holonic control architectures in *section 2.2.1* as well as multi-agent architectures for building the informational part of holon in *section 2.2.2*. Due to the composition characteristic of communicating material and the high flexibility requirement throughout its whole lifecycle, recursive architectures are reviewed in *section 2.2.3* to find or design a recursive architecture for the communicating concrete application. At the end, a synthesis is given in *section 2.2.4*.

### 2.2.1 Holonic Control Architecture (HCA)

In the manufacturing field, the flexibility of the control system has always been required to adapt to environmental changes. One of the first holonic manufacturing architecture, PROSA (Product-Resource-Order-Staff) is proposed by Van Brussel (Van Brussel *et al.*, 1998). This architecture consists of three types of basic holons: *order* holons, *product* holons, and *resource* holons. The general architecture is shown in Figure 34. The authors introduced *staff* holons which can be added to assist the basic holons to complete their work. Holons can cooperate to achieve adaptation and flexibility requirements.

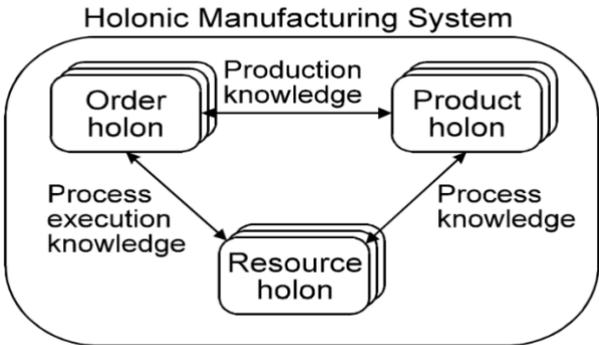


Figure 34 PROSA architecture (Van Brussel *et al.*, 1998)

Holons are structured by using object-oriented concepts. The *resource* holon contains a physical part which is production resource and an informational part that controls the resource; The *product* holon holds the process and product knowledge; The *order* holon assigns tasks to other holons to ensure the production.

These holons cooperate by exchanging the manufacturing system knowledge. The *order* holons communicate production knowledge with the *product* holons, which can then exchange process knowledge with the *resource* holons to produce the required order. Meanwhile, the *resource* holon shares the process execution knowledge with the *order* holon. Based on the process execution knowledge, the *order* holon can ensure the production on time. The *staff* holon has an overview of all resources and all orders, it generates a schedule as advice to the *order* holons and *resource* holons. These basic holons can follow the advice when there are no disturbances. However, once the disturbances or changes occur in the system, the basic holons can ignore the advices from the *staff* holons and make their own decisions.

ADACOR (ADaptive holonic COntrol aRchitecture) is a nature-inspired holonic architecture proposed by (Leitão and Restivo, 2006). A pheromone-like propagation mechanism is used to distribute information when disturbances occur. In this architecture, four holons are defined: *product holon*, *task holon*, *operational holon* and *supervisor holon*. These holons are structured in a decentralized control architecture to adapt to environmental changes. The defined holons are quite similar from PROSA. *Product*, *task*, and *operational* holons have the same functionalities as the *product*, *order* and *resource* holons in PROSA. These holons cooperate via an architecture as shown in Figure 35. In this approach, the *supervisor* holon acts an important role because it is responsible for forming and coordinating groups of holons.

In this architecture, holons are organized in a hierarchical structure with four levels: planning, coordination and operational level. The *supervisor* holons propose optimized execution plan to the *task* holons and monitor the execution of operating holons. When disturbance occurs, *operational* holons propagate the need for reorganization via the supervisor holon. The neighbor *operational* holons sense the pheromone and re-organize themselves into a heterarchical structure. Holons can then interact directly with the *task* holons to achieve an alternative schedule plan that minimizes the impact of the unexpected disturbance.

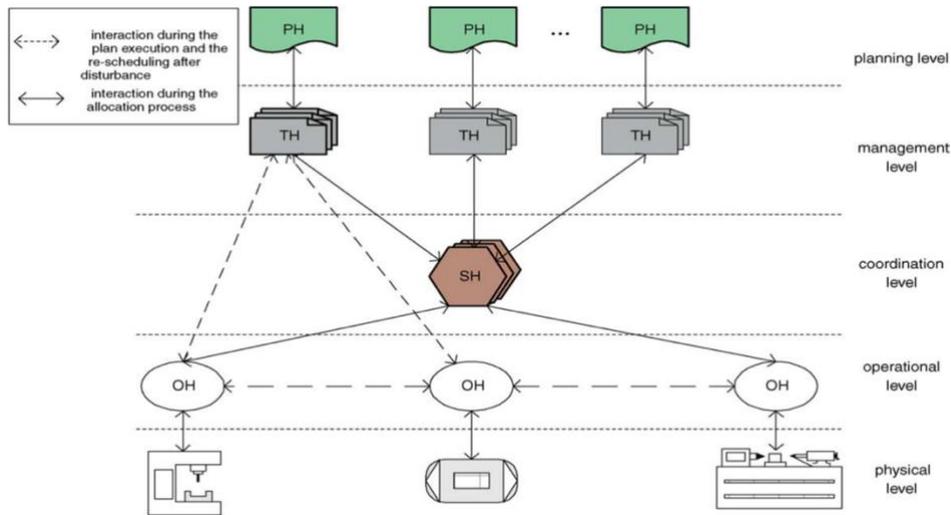


Figure 35 ADACOR architecture (Leitão and Restivo, 2006)

Pujo et al. propose an isoarchy structure named PROSIS (Product, Resource, Order, Simulation for Isoarchy Structure) (Pujo et al., 2009). This architecture inherits from PROSA. It uses the concept of isoarchy to define a flat holonic form where holons are in a unified level. This decentralized architecture is designed to increase the productivity, the flexibility and the responsiveness of the production system. In this unique decision level structure, the staff holon is replaced by a Simulation Holon, which is designed to evaluate the prospects for evolution of the production system and prevent emergences.

These holonic control architectures present the approaches to manage the resource of a product (physical part) and the information (digital part). Holons are organized in hierarchies, which can evolve according to certain mechanisms. This ability confers to these architectures some adaptation features. To realize the informational part, Multi-Agent Systems are generally used for the management of the digital part.

### 2.2.2 Multi-Agent System (MAS)

In the past two decades, multi-agent system has attracted a lot of attention. Agents can work independently and cooperate to resolve complex tasks. Due to its high flexibility, the agent paradigm has been widely applied in different areas (Sardouk et al., 2013) (Dorri et al., 2018).

According to Dorri (Dorri et al., 2018), there are seven different types of organizations as shown in Figure 36: Flat (a), Hierarchical (b), Holonic (c), Coalition (d), Team (e), and Matrix (f) and

Congregation (g). The flat is the most basic organization structure, in which all agents can communicate with their neighbors. The agents with common objectives can directly cooperate. In a hierarchical organization, agents are grouped in a tree-like structure, leaf agents can communicate with others via their parent agent (Ma and Zhang, 2014). As the parent agent has knowledge of its children, a local optimal decision can be provided. Meanwhile, the root agent which has a global view could provide optimal solution to face environmental changes. In holonic organization, the agents are considered as the informational part of holons as in (Esmaeili et al., 2017). The holons with the same interest or with same features can be grouped into a higher layer holon. Agents with the same parent can communicate with each other.

In the four other kinds of organization (Team, Coalition, Congregation and Matrix), agents are temporarily or permanently gathered into groups based on their goal. For example, an agent can be part of more than one group in coalition structure (Manisterski et al., 2007), and the different groups communicate with the others via a flat structure to achieve a common goal.

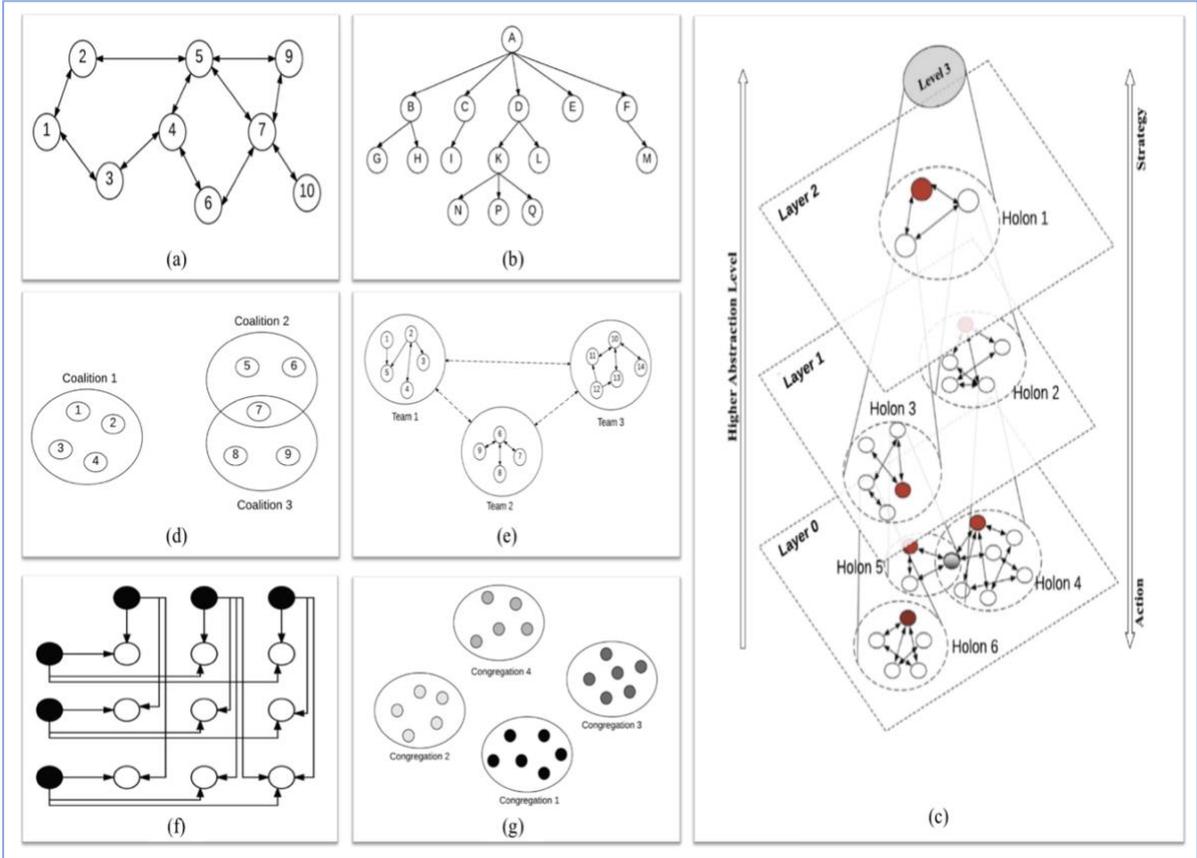


Figure 36 Organization approach of agents: Flat (a), Hierarchical (b), Holonic (c), Coalition (d), Team (e), Matrix (f) and Congregation (g) (Dorri et al., 2018)

In matrix organization, each agent is controlled by at least two other agents. This multiple supervision could reduce the error rate during production (Meszyński and Sokolov, 2017). It is inspired from human society where a person can be guided by a supervisor and a team leader. However, the mentioned structures define relative static cooperation between agents but only one (the holonic organization) facilitates the dynamicity of the structure and the use of recursive patterns, which are mandatory for communicating concrete application. In the Holonic or MAS communities, several attempts were made to address recursion in different ways. The recursive solutions are reviewed in the following section.

### 2.2.3 Recursive holarchy or MAS

(Le Mortellec *et al.*, 2013) presents SURFER (SURveillance active FERroviaire, translated as “active train monitoring”), an embedded holonic fault diagnosis architecture for complex transportation systems (CTS) where train, vehicles and doors are considered as intelligent products. The generic holonic architecture was presented in Figure 37. All elements are represented by Holons.

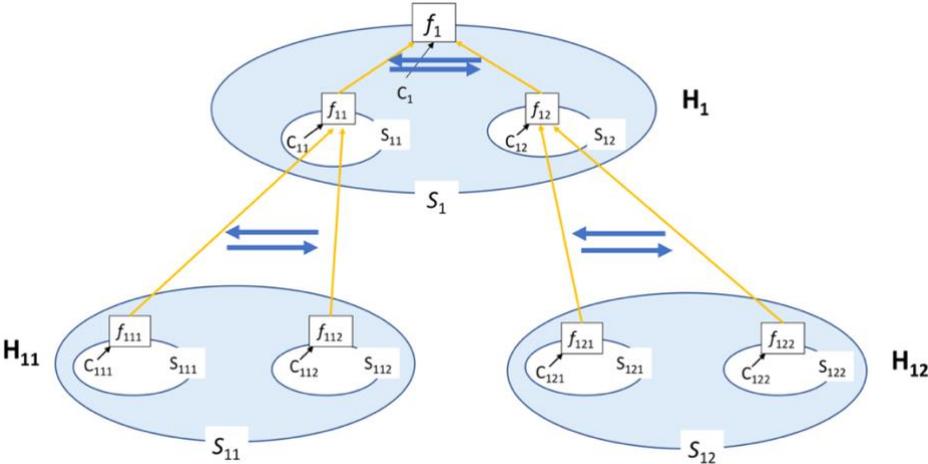


Figure 37 Holonic architectural model for embedded diagnosis

In this architecture,  $H_1$  represents a train. It has two children (vehicles  $H_{11}$  and  $H_{12}$ ). Besides, each vehicle also has two doors. The system  $S_1$  is decomposed into subsystem  $S_{11}$  and  $S_{12}$ ,  $S_{112}$  is a sub-system of  $S_{11}$ . In this hierarchical structure, each level takes the results from lower layer into account and reports results to upper layer. This approach presents the information exchanges on the same level and the recursion for different levels. This work describes a modeling framework but does not detail the composition and decomposition mechanisms needed to manage a dynamic system.

Since the informational part of holon is always realized by agents, recursive agent architecture can be considered for the digital part of our application. A recursive agent meta model (RAMM) was proposed by (Vo et al., 2012) for multi-scale modeling (MSM). This meta-model has four main parts: species, agent, environment, and scheduler. Agents of different species can form a hierarchical structure. For example, a root agent located at level 0 consists of two agent species on level 1 (species 1 and species 2). At level 2, species 3 is a micro species of species 1 on the level 1. The authors define the capture and release methods to add or remove micro species. However, the interactions between agents are not detailed.

Abdoos et al. use holonic multi-agent system to model a large traffic network (Abdoos et al., 2013). Different intersection controllers and regions are represented by holons in a two-level hierarchical architecture. Each intersection controllers are represented by a holon in the first level, and region control in the second level. The authors propose the holonic Q-learning method which helps agents to decide their next action based on states to analyze the signal control. Holons in the lower level send the information from the environment to the holons in the higher level, which estimate and provide the actions. The experimental results shows that this method can reduce the average delay. But the aggregation and decomposition of holons are not detailed in this method.

(Sallez et al., 2015) present a recursive holarchy in logistics for Physical-Internet using (PI)-containers. These containers have some communicational and decisional capabilities. According to their size and function, three types are defined from the largest to the smallest: transport container, handling container and packaging container. Encapsulation and composition concepts are defined for each kind of container. Indeed, containers of the same type can be composed as a “composite” PI-container. Small containers (handling and packaging) can be encapsulated in a bigger one. However, the compositional mechanisms are not formally specified here too.

In Esmaeili et al. (Esmaeili *et al.*, 2017), recursive holarchies are used to build and manage holonic structures for large-scale MAS (LS-MAS). The four-level holarchy representation is illustrated as in Figure 38 where the heads of holons are shown in red. The holons can communicate with the ones in the same level, or with holons in upper or lower levels through the heads of the holons. The authors detailed the location information-based algorithm to construct the holarchy, the head selection algorithm among holons, and the reorganization

mechanism for the architecture. This model can be used well with large-scale multi-agent systems. Task assignment application is evaluated to prove the usefulness of recursive architecture.

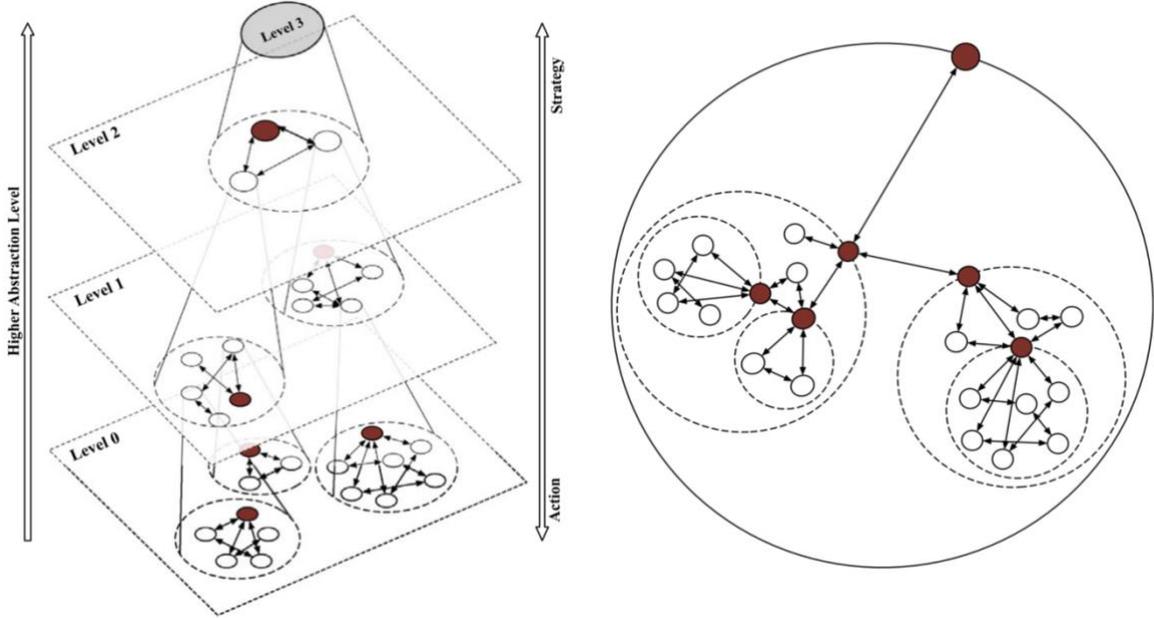


Figure 38 Social-based holonic architecture (Esmaeili et al., 2017)

Occello et al. present a generic Recursive Multi-Agent System (MAS-R) for supervision and observation of WSN (Ocelllo et al., 2019). The proposed MAS-R agent model is composed by four parts: Knowledge (*K*), Observation (*O*), Composition (*C*) and Recursive Interaction (*RI*). These four parts act their unique role in the multi-agent system. The *K* part records the basic knowledge and abstraction level of agent. The *O* part is responsible for observing the state of composition. Composition tasks are performed in the *C* part. The *RI* part of an agent updates its own data, processes, and transfers data to the other agents.

The hierarchical MAS-R structure is illustrated in Figure 39. There are three levels. At the level 0, agents  $a_0^i$  are “Elementary” agents which have a physical applicative part and a recursive agent part. The agent  $a_i^j$  represents the agent  $j$  on the  $i^{th}$  level. These agents on high levels ( $i$  is greater than zero) are called “Partial” agent. A “Complete Elementary” agent is composed by its elementary agent and all theirs directly related “Partial” agents (from abstraction levels). “Composed” agents are the gathering of few “Partial” agents from the same abstraction level.

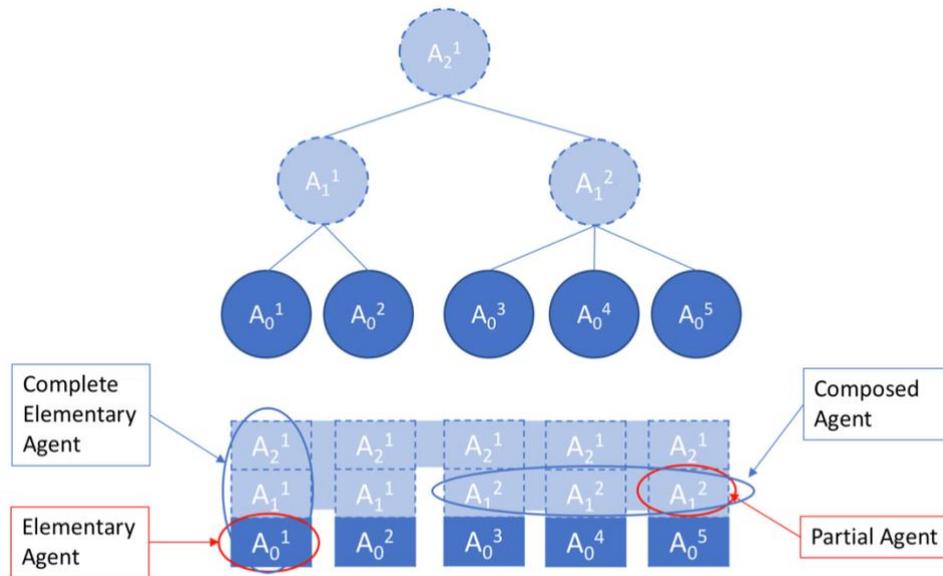


Figure 39 The MAS-R architecture

The authors presented four types of transformations functions for each component of a MAS (Agent, Interactions, Organization and Environment) respectively  $PA$ ,  $PI$ ,  $PE$  and  $PO$ :

- $PA$  is an operator grouping agents from one level and associating them to an agent of the upper level;
- $PE$  is an operator grouping elements of the environment from one level into an element of the environment from the upper level;
- $PI$  is an operator transforming interactions from agents of a level to interactions to the upper level. Perceptions/actions and messages of agents of a level  $N$  are transformed thanks to  $PI$  to perceptions/actions and messages of agents of level  $N+1$ .
- $PO$  is an operator that transforms relations between agents. A relation between agents of level  $N$  grouped in different agents of level  $N+1$  is transformed by  $PO$  into a relation between agents of level  $N+1$ .

Besides, the authors also detailed the composition mechanism for this recursive system. Once composition conditions are met, a composed parent agent can be created and other agents are invited to join it.

Khelifa and Laouar propose a holonic intelligent decision support system for urban project planning (UPP) (Khelifa and Laouar, 2020). The proposed three-level holonic architecture is shown in Figure 40. At the top level, the project holon, urban area holon and the authority holon share the same resource. The authority holons records the authority relations between the area and its sub-area, the project and the sub-project. These three holons are structured in a

recursive way. For example, the area territory holons can be divided into areas in the middle level and sub-areas in the low level. The authors use multi-objective ACO (Ant colony) algorithm to optimize the urban planning from the bottom level to the top level (from the sub-area to the Area Territory). Besides, a H-MACO (Holonc multi-objective ant colony optimization) algorithm is proposed to improve the global urban planning with the global objectives. This method can achieve an efficient solution for large scale applications where the network is static. Nevertheless, the reconstruction of network is not discussed.

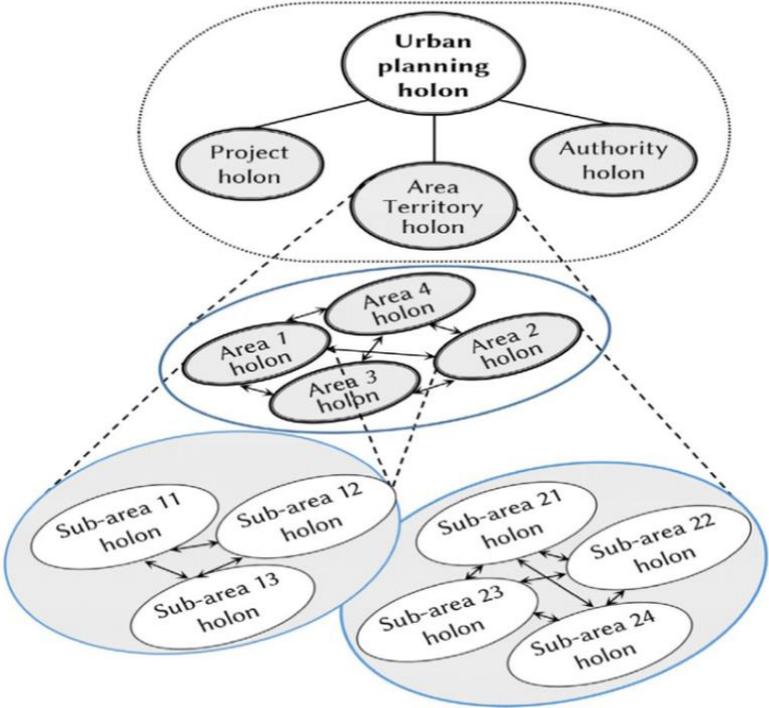


Figure 40 Three-level holonic architecture for urban planning (Khelifa and Laouar, 2020)

Nurdin and Muchallil propose a Rol-Interaction-Organization (RIO) model based holonic multi-agent system for the control system in microgrid (Nurdin and Muchallil, 2019). The holonic architecture is present in Figure 41. There are four types of roles in this structure: the *part*, the *head*, *multi-part* and the *StandAlone*. A *part* holon cannot directly communicate with holons of other groups; A *head* of a group is responsible for collecting and disseminating information to the group members; a *multi-part* holon is the one in two different groups; the *StandAlone* is the holon which can communicate with the holon outside the groups. Although the authors present the satisfaction module to define the role of agents, the selection of head in a group and the interactions between holons are not specified.

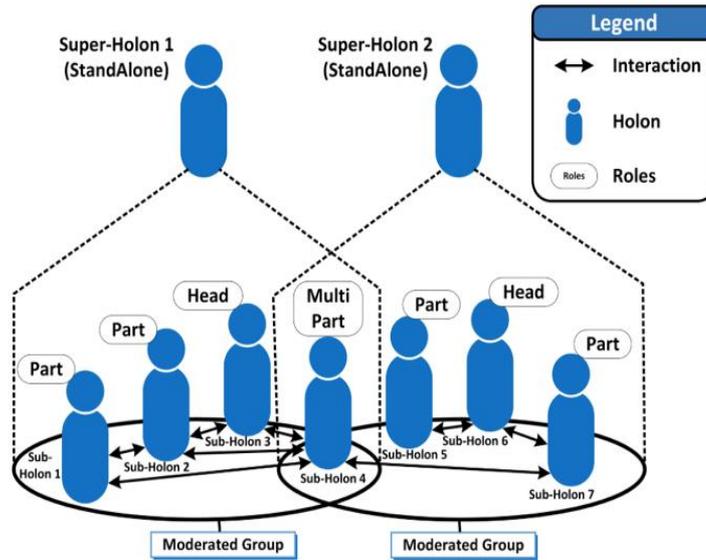


Figure 41 Holonic architecture for microgrid (Nurdin and Muchallil, 2019)

The concept of recursion has been discussed or applied in mentioned works. A summary is given in Table II. As we try to define a dynamic recursive architecture for CM applications, four different characteristics are used to analyze the presented works: application domain, architecture, formalization of agents and aggregation (or composition) mechanism. Application domain presents their applied field. The architecture records whether the approach can evolve with time or not (static or dynamic). Whether they proposed an agent model or composition mechanisms are recorded in formalization of agent and aggregation mechanism respectively.

From Table II, we can see that the need of recursion is expressed in different applicative areas. The WSN application is closest to our needs and may provide an adapted architecture. The approach given by (Le Mortellec *et al.*, 2013) is applied for static systems as well as the proposal of (Esmaili *et al.*, 2017), and (Khelifa and Laouar, 2020). These approaches are not adapted for dynamic systems, such as the communicating concrete application. Although the aggregation notions (encapsulation and composition) and the interactions between agents are given by (Sallez *et al.*, 2015), the authors do not detail the aggregation mechanism. (Ocelllo *et al.*, 2019) presents a recursive agent model in MAS-R, and details the interactions and aggregation mechanisms for a dynamic architecture. This method seems to be suitable for CM applications. Thus, this approach is a good candidate for application to the CM context.

Table II. Summary table of recursive MAS

Reference	Application domain	Architecture	Formalization of agent	Aggregation mechanism
(Le Mortellec <i>et al.</i> , 2013)	CTS	Static	No	No
(Abdoos <i>et al.</i> , 2013)	Traffic	Static	No	No
(Sallez <i>et al.</i> , 2015)	Logistics	Dynamic	No	Yes
(Esmaeili <i>et al.</i> , 2017)	LSMAS	Static	Yes	No
(Ocelllo <i>et al.</i> , 2019)	WSN	Dynamic	Yes	Yes
(Khelifa and Laouar, 2020)	UPP	Static	No	Yes
(Nurdin and Muchallil, 2019)	Microgrid	Static	No	No

Agents and holons have much common properties. They can both react independently or cooperate with others to adapt to environmental changes. There are still some differences between holons and agents. One of the most important is that agents have not internal separation between informational and physical part. Barbat *et al.* (Bărbat *et al.*, 2001) claim that holonic approach overcomes the agent limitations in the field of forming multi-level hierarchies or creating temporary structures in dynamic environment. The similar opinion was expressed by Leitão (Leitão and Karnouskos, 2015), who claimed that “the holonic paradigm, extended by agent norms and policies, offers a starting point for MAS-based systems that also support hierarchical (control) structures and decision making”. Vrba *et al.* (Vrba *et al.*, 2011) go further and use the term “holonic agent”.

### 2.3 Synthesis

In this chapter, energy modeling and saving mechanisms to extend WSN lifetime in the physical part and recursive architecture for the management in digital part are both reviewed. In the physical part, three aspects are especially pointed out: collection structure for data gathering, time synchronization protocols for optimizing radio sleep mode and aggregation function for

reducing the transmitted data. As these three aspects are important for energy saving during data collection, they should be considered for constructing the analytical energy consumption model. This model should provide an approximate and rapid remaining energy information for the usage and maintaining of the CM products. Therefore, in the following chapter, an analytical energy consumption model for a simple chain structure is proposed in *section 3.2* considering data collection without aggregation and with aggregation in *section 3.2.1* and *section 3.2.2* respectively. Besides, a general analytical model is proposed in *section 3.3* for the every kind of hierarchical WSN structures, considering lossy or lossless aggregation.

To overcome the limitation of processing power and storage capability at nodes, the intelligent holonic control system is reviewed for the management of the digital part. To represent the physical part in the digital world, different MAS approaches are analyzed to find solution compatible with our requirements. Due to the composition and decomposition characteristics, a recursive architecture is needed for the communicating concrete application. Since the MAS-R architecture seems to meet our needs, an extension of this approach to the CM context is proposed in the *Chapter 4*. The recursive architecture and the adapted agent model are presented in *section 4.1* and *section 4.2* respectively. The composition mechanism, interaction between agents, the cooperation between aggregated elements as well as the application for lifetime estimation are detailed from *section 4.3* to *section 4.6*.



# Chapter 3

## Energy consumption for periodic data collection in WSN

### Contents

<a href="#">3.1 Energy model of a communicating node</a> .....	62
<a href="#">3.2 Data collection in a chain communication structure</a> .....	64
<a href="#">3.2.1 Analytical model for data collection without aggregation</a> .....	68
<a href="#">3.2.2 Analytical model for Data collection with aggregation</a> .....	71
<a href="#">3.3 Analytical model for data aggregation in WSN</a> .....	73
<a href="#">3.3.1 Energy consumption estimation method in WSN</a> .....	73
<a href="#">3.3.2 Model for tree structure</a> .....	79
<a href="#">3.3.3 Model for other structures</a> .....	82
<a href="#">3.4 Conclusion</a> .....	83

This chapter presents the proposed analytical energy consumption models for periodical data collection in WSN. It begins with the energy model of a communicating node in *section 3.1*. Data collection process is detailed in *section 3.2*. The position-based analytical energy models for data collection without data aggregation in a chain structure are presented in *section 3.2.1* and with aggregation in *section 3.2.2*. Based on the chain structure model, a generic energy consumption model for every kind of WSN structure is given in *section 3.3*. Then, a conclusion is presented in *section 3.4*.

### 3.1 Energy model of a communicating node

During data collection, communication takes a large proportion of energy consumption. The radio communication modules are responsible for sending and receiving data. As mentioned in *Chapter 2*, distance-based energy model cannot be applied for communicating concrete applications. Therefore, a power-based energy model is used, where four different states are considered for the radio communication module:

- Transmit ( $T_x$ ): transmitting message to other nodes
- Receive ( $R_x$ ): receiving message from other nodes
- Idle: active but not receiving or transmitting data
- Sleep: inactive with a low energy cost

Radio modules can shift into different states with some delays as depicted in Figure 42. Transition delays depend on the radio module device and especially on its host microcontroller. These transition costs have been studied (Zhou *et al.*, 2011). Compared with active duration for transmission and data processing, these delays are small (some microseconds). Therefore, these delays are considered as negligible and thus are ignored in our model. As a consequence, to give an approximate WSN lifetime, our model only takes into account the active and inactive states of the radio module.

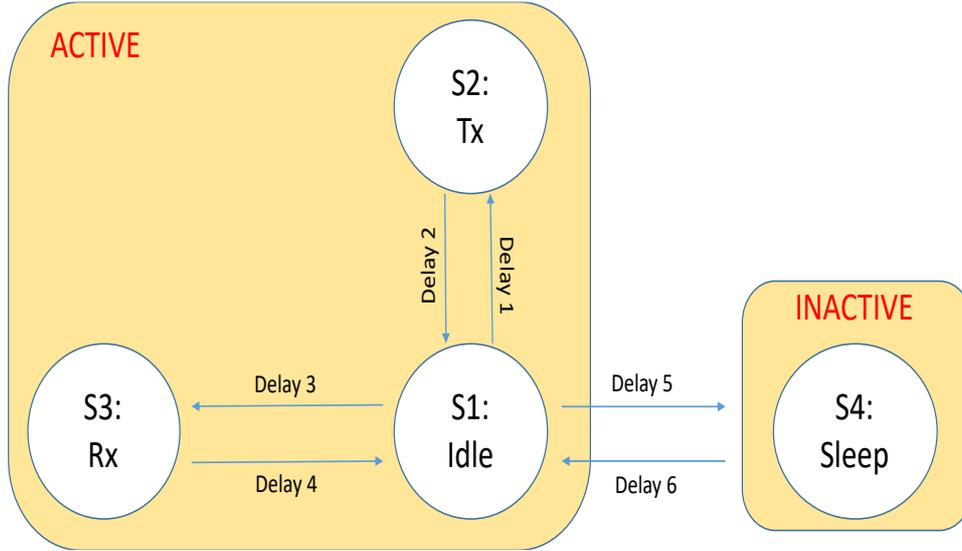


Figure 42 different states of radio module

The energy consumption of the radio communication module can be simplified with the active and sleep states as in (10):

$$\begin{aligned}
 E &= E_{Tx} + E_{Rx} + E_{Idle} + E_{Sleep} \\
 &= E_{active} + E_{Sleep} \\
 &= P_{active}D_{active} + P_{Sleep}D_{Sleep}
 \end{aligned} \tag{10}$$

where  $P_{active}$  is the mean power of the three active states ( $Tx$ ,  $Rx$ ,  $Idle$ ) and  $P_{sleep}$  is the power spent in sleep state.  $D_{active}$  and  $D_{sleep}$  correspond to the duration in the active and sleep states respectively, mandatory to estimate the energy cost of data collection per round. All parameters used for the analytical energy consumption model are defined in Table III.

Table III. Proposed model's symbols

Notation	Unit	Definition
$E_i$	$J$	Communication energy consumption of node $i$
$E_{total}$	$J$	Total energy consumption of network
$P_{active}$	W/s	Power of active state
$P_{sleep}$	W/s	Power of sleep state
$D_{active}(i)$	$s$	Active duration of node $i$
$D_{sleep}(i)$	$s$	Sleep duration of node $i$
$D_{tx}(i)$	$s$	Duration for message transmission of node $i$
$D_{rx}(i)$	$s$	Duration for message reception of node $i$
$S_{min}$	byte	Minimal size of transmission data packet
$S_{max}$	byte	Maximal size of transmission data packet
$\alpha$	$s$	Transmission duration
$\alpha_{min}$	$s$	Transmission duration for data packet of minimal size
$\alpha_{max}$	$s$	Transmission duration for data packet of maximal size
$\beta$	$s$	Inter Frame Space time
$\gamma$	$s$	Time for ACK message
$d_{syn}$	$s$	Synchronization duration
$T$	$s$	Data collection period
$N$		Chain size (Number of nodes in a chain)

### 3.2 Data collection in a chain communication structure

In this section, we model a collection process that is as generic as possible, in order to be applied to different technologies. The collection begins from the furthest node. Its workflow is illustrated in Figure 43. At the beginning, its communication module is activated to transfer messages to its parent node. This node then waits an acknowledgment (*ACK*) message from its parent node before letting its communication module enter into sleep mode. A delay is used to ensure the reception of an *ACK* message from its parent node. Its message will be sent again if the waiting time exceeds the prefixed delay. This node periodically reactivates its communication module for a new transmission process.

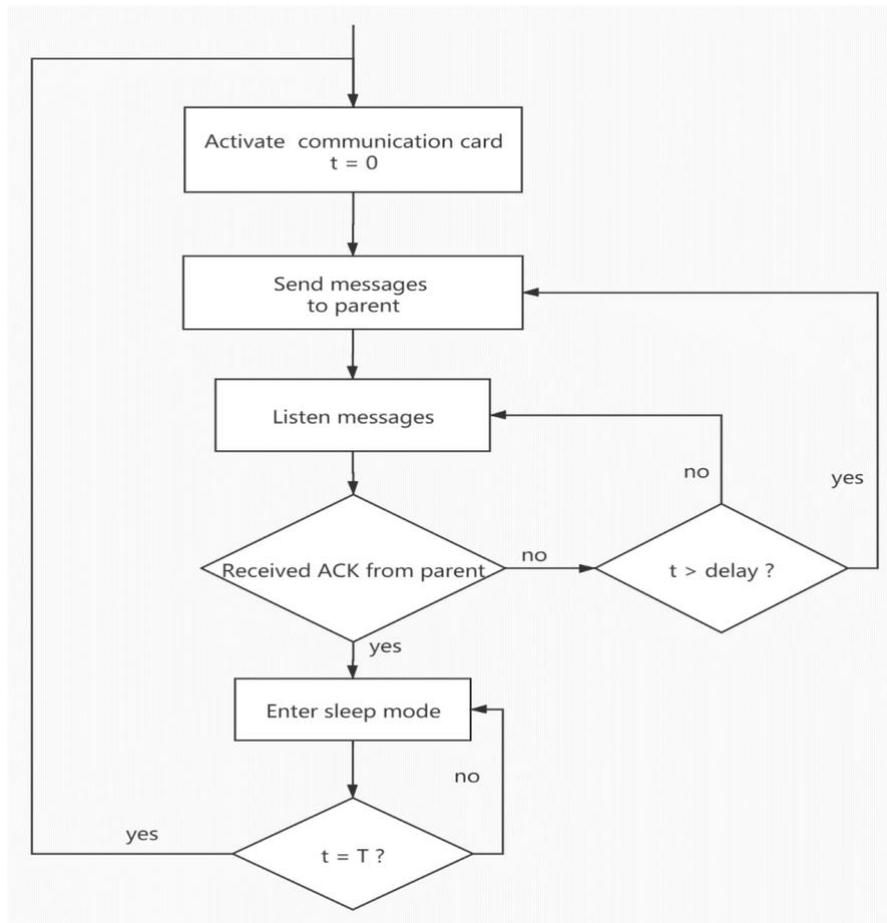


Figure 43 Process at the furthest node

Data aggregation functions can be performed at intermediate nodes. For data collection without aggregation, an intermediate node will first transfer all received messages. It then sends an *ACK* message to its child node which can then set their communication module to sleep mode. After replying an *ACK* message, an intermediate node then transmits its own message to its dedicated parent node and waits for the response.

As described in *section 2.1.1*, using data aggregation techniques can reduce the size of transferred messages and extend network lifetime. The flowchart of intermediate nodes for data collection with aggregation is illustrated in Figure 44. At the beginning, an intermediate node activates its communication module to listen to messages from its child nodes. Instead of directly transmitting received messages to its parent node, it first stores the messages. Once all messages are received, it sends an *ACK* message back and aggregates received messages with its own one. Only the aggregated message is then sent to its parent node. Aggregated message is sent again if the waiting time exceeds a given delay. Once it received an *ACK* message from its parent, a node then sets its communicating module to sleep mode to save energy. The

communication module is reactivated when its local time is up to the scheduled time for the following iteration. To ensure nodes can wake up at the right moment, the facing challenge is synchronization.

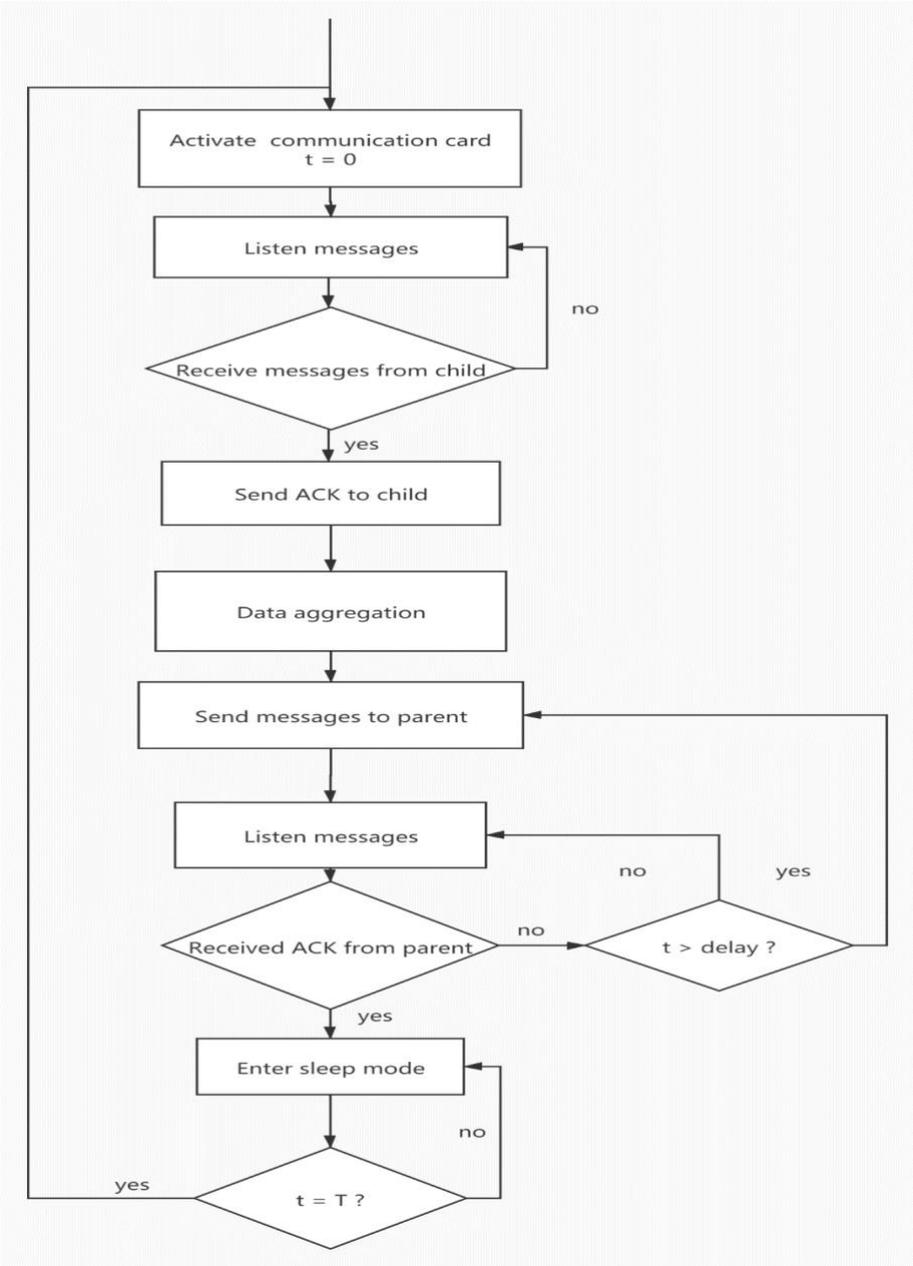


Figure 44 Process at intermediate node

The different processes can be expressed via chronograms. The chronogram on Figure 45 depicts the radio activity of the first two nodes in a chain structure without data aggregation. In a chain-based structure, and more generally in a wireless environment, a node can receive messages from other nodes within its communication range. To avoid mixing data, inter-frame spaces (IFS) are generally used in MAC protocols to shift the communication messages. In the case of periodic data collection, synchronization is performed during the first iteration period

(SYNC). All nodes send messages with the same period  $T$ . For a node  $I$  in the  $j^{\text{th}}$  iteration, the  $t_{ij\_begin}$  represents its start time for data collection,  $t_{ij\_sleep}$  records the time when it turns off the radio communication module (i.e. entering into sleep mode). At the beginning of data collection process, all nodes are awake.

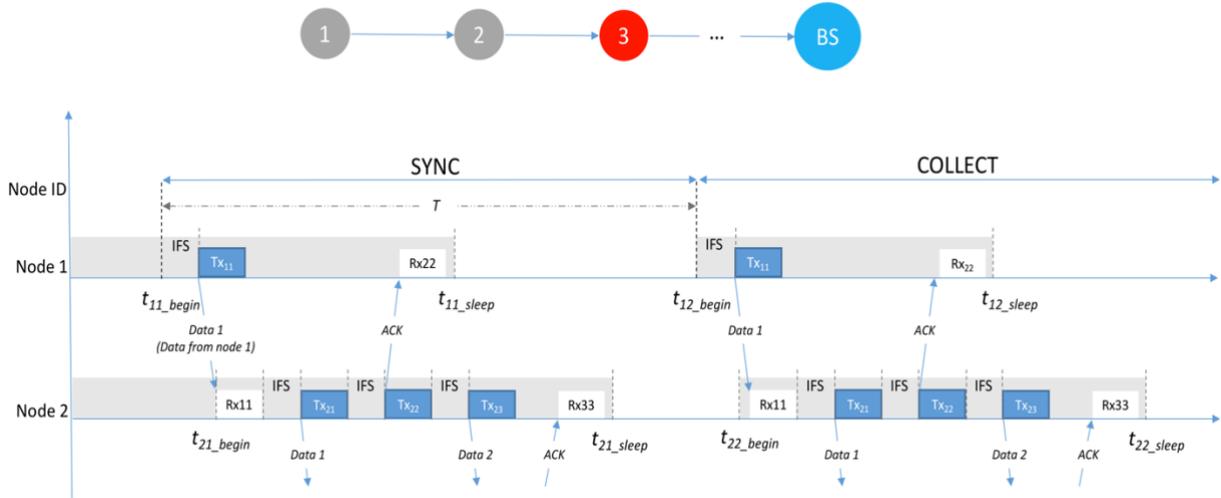


Figure 45 Chronogram of radio module's activities without aggregation

In the chronogram, different blocks are used to represent activities of communication modules. The modules are in *sending* or *receiving* states with  $Tx$  or  $Rx$  respectively. The transmission activities are represented by the node  $ID$  and its transfer message number. For example,  $Tx_{ij}$  is the  $j^{\text{th}}$  message transfer from node  $i$ . Meanwhile, the reception activity is presented with the sender  $ID$  and the message number (like  $Rx_{11}$  is the message 1 from node 1). The modules are in *idle* state (grey color) when they are active but not sending or receiving data.

Synchronization process begins with the furthest node (here, node 1) at the start time  $t_{11\_begin}$ . The time before turning off communication module is recorded as  $t_{11\_sleep}$  after it received an  $ACK$  message from its parent (node 2). It will turn communication module on for next iteration at time  $t = t_{12\_begin}$  where  $t_{12\_begin} - t_{11\_begin} = T$ . The total active duration of the furthest node can then be computed with the recorded start time  $t_{11\_begin}$  and sleep time  $t_{11\_sleep}$ .

Unlike the furthest node, start time at intermediate nodes (here node 2) depends on the previous node. It records the message arrival time of node 1 as its start time  $t_{21\_begin}$ . For data collection without aggregation, it first transmits all messages from its child (node 1) and then sends its own message to its parent (node 3). Once receiving  $ACK$  message from its parent, it records the time  $t_{21\_sleep}$  and lets its communicating module enter in sleep mode. For next iteration, it will

reactive the communication module a little earlier than its child to ensure the reception of messages. Although using sleep mode can reduce energy consumption at nodes, the hot-spot problem in chain structure is not solved: the closer the node is to the BS, the more energy the node consumes. Therefore, using data aggregation technique could be suitable. It can not only balance energy consumption in WSN, but also extend lifetime. Instead of transmitting all messages from child, only the aggregated message is sent for all intermediate nodes in the case of data collection with aggregation. The associated chronogram is shown in Figure 46.

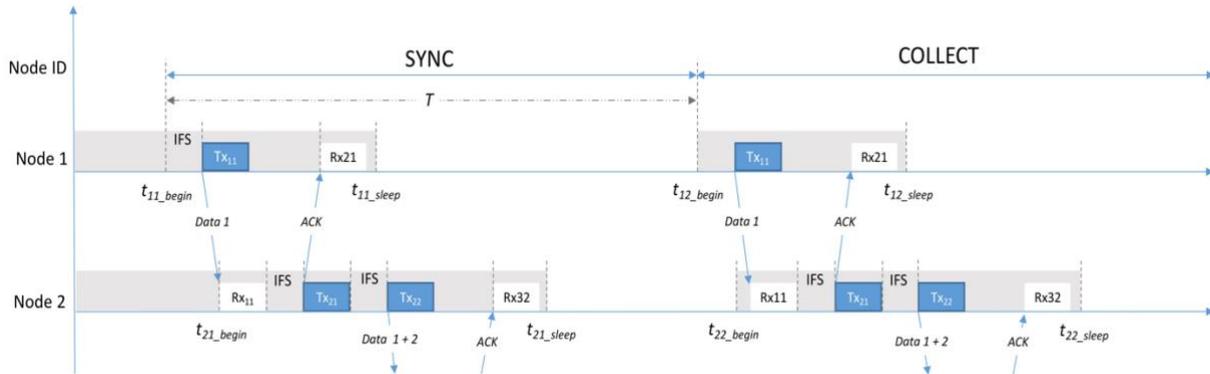


Figure 46 Chronogram of radio module's activities with aggregation

The process of the furthest node is the same. However, intermediate nodes first stores messages from their child and replies with an ACK message. The received messages are then aggregated with their own message. Only the aggregated message will be sent to parent node. In this case, the activity duration ( $t_{ij\_sleep} - t_{ij\_begin}$ ) for communication at all nodes is less than that in data collection without aggregation. Duty cycle at these nodes is reduced, and lifetime is increased.

### 3.2.1 Analytical model for data collection without aggregation

Since lifetime is the most important performance metrics for our application, knowing the residual energy of nodes is necessary for energy management. Building a theoretical energy consumption model could help us to know how the energy is consumed. As a consequence, an energy consumption estimation model, based on the mentioned data collection process, is presented hereafter.

Energy consumption of nodes depends on their activities. In this study, the power on active mode and sleep mode are  $P_{active}$  and  $P_{sleep}$  respectively.  $T$  is the period between two cycles of data collection. For building our model and in order to simplify those variables, we replace the

mentioned transmission  $Tx$ , durations of IFS, time for sending or receiving  $ACK$  message and the time needed for the node synchronization (i.e. node should wake up a little bit earlier before receiving messages from a child node) by some symbols:  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $d_{syn}$  (see Figure 47). In this section, we suppose that the size of  $ACK$  message is the same as one transmission message, therefore,  $\alpha$  is equal to  $\gamma$ .

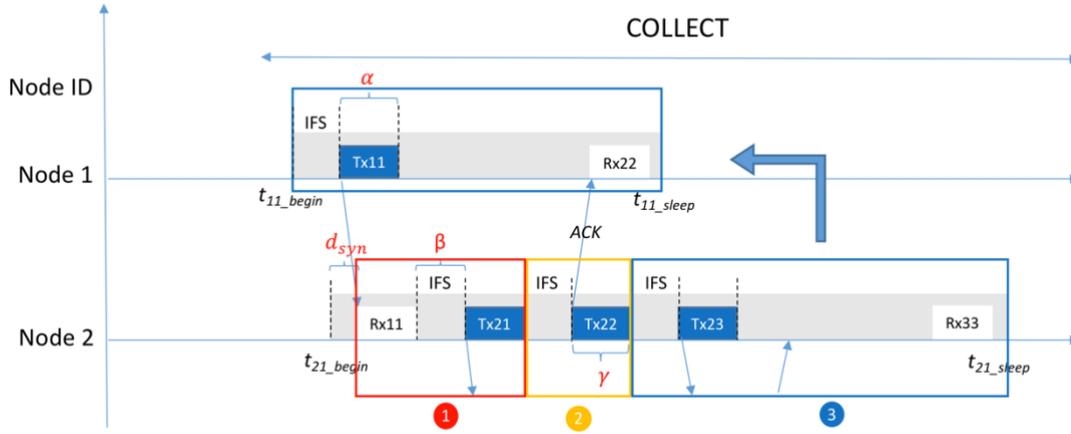


Figure 47 Explanation of model parameters

The node 1 first sends its message  $Tx_{11}$  ( $\alpha$ ) after an IFS ( $\beta$ ) to ensure the channel is free. It then waits an  $ACK$ . The node 2 retransfers received message ( $\alpha + \beta$ ) and then replies an  $ACK$  message to node 1 ( $\alpha + \beta$ ). After receiving  $ACK$  from node 2, node 1 sets its radio module to sleep mode. Meanwhile, it records the sleep time to compute the beginning time for next iteration. To summary, the total duration of node 1 during the data collection phase is  $3\alpha + 3\beta$ , it is the minimal time for sending message and getting a reply from parent node (blue block in Figure 47).

For intermediate nodes (such as node 2 in the example) in a  $N$ -node chain structure, their active duration consists of three parts as the red, orange, and blue blocks shown in Figure 47: (1) transfer the received messages ( $2\alpha + \beta$ ); (2) send an  $ACK$  message to its child ( $\alpha + \beta$ ); (3) Send its own message and wait for an  $ACK$  like the node 1 ( $3\alpha + 3\beta$ ). Besides, node 2 should wake up a little bit earlier than its previous nodes with  $d_{syn}$  for the phase of data collection. Therefore, its total active duration is:  $\alpha + 2 * (\alpha + \beta) + 3\alpha + 3\beta + d_{syn}$ . The process for all intermediate nodes is the same. The closer to the BS, the more information needs to be transmitted. The equation can be generalized for the  $i^{th}$  node. The minimum activity duration of intermediate node  $i$  is  $\alpha + (i - 1) * 2 * (\alpha + \beta) + 3\alpha + 3\beta + d_{syn}$ .

The last node ( $N^{th}$ ) of a chain works as intermediate nodes for the first two steps (activity duration is  $\alpha + (i - 1) * 2 * (\alpha + \beta)$ ). But its last step takes less time: once received its message ( $\alpha + \beta$ ), the BS will reply with an *ACK* message ( $\alpha + \beta$ ). Therefore, the active duration of node  $i$  in a  $N$ -node ( $N \geq 3$ ) chain structure can be expressed in (11).

$$D_{active}(i) = \begin{cases} 3\alpha + 3\beta, & \text{when } i = 1 \\ (i - 1) * 2 * (\alpha + \beta) + 4\alpha + 3\beta + d_{syn}, & \text{when } 2 \leq i \leq n - 1 \\ (i - 1) * 2 * (\alpha + \beta) + 3\alpha + 2\beta + d_{syn}, & \text{when } i = n \end{cases} \quad (11)$$

Where  $\alpha, \beta$  and  $\gamma$  values are constant.  $(3\alpha + 3\beta)$ ;  $(4\alpha + 3\beta + d_{syn})$  and  $(3\alpha + 2\beta + d_{syn})$  can be respectively replaced by constant values  $k_1, k_2, k_3$ . The equation can then be simplified as follows:

$$D_{active}(i) = \begin{cases} k_1, & \text{when } i = 1 \\ (i - 1) * 2 * (\alpha + \beta) + k_2, & \text{when } 2 \leq i \leq n - 1 \\ (i - 1) * 2 * (\alpha + \beta) + k_3, & \text{when } i = n \end{cases}$$

The data collection is performed periodically (cycle collection period  $T$ ). With the duration of activity above, the **duration of sleep mode** for node  $i$  in a  $N$ -node chain structure can then be computed as in (12).

$$D_{sleep}(i) = T - D_{active}(i) \quad (12)$$

In our study, we only consider the active and sleep mode, where the power consumption can be presented by  $P_{sleep}$  and  $P_{active}$  for sleep and active states respectively. **Energy consumption per round for node  $i$  in  $N$ -node chain structure** can then be estimated with their duration. The energy consumption  $E_i$  of the node  $i$  can be calculated as in (13):

$$E_i = D_{active}(i) * P_{active} + D_{sleep}(i) * P_{sleep} \quad (13)$$

Knowing the energy consumption of all nodes, it is then possible to estimate energy cost of whole network for each round. The total cost for one round can be calculated as in (14):

$$\begin{aligned} E_{total} &= \sum_{i=1}^n E_i = \sum_{i=1}^n (D_{active}(i) * P_{active} + D_{sleep}(i) * P_{sleep}) \\ &= \sum_{i=1}^n (D_{active}(i) * P_{active} + (T - D_{active}(i)) * P_{sleep}) \\ &= \sum_{i=1}^n (D_{active}(i) * (P_{active} - P_{sleep}) + T * P_{sleep}) \\ &= n * T * P_{sleep} + \sum_{i=1}^n D_{active}(i) * (P_{active} - P_{sleep}) \end{aligned} \quad (14)$$

The total active duration is computed as in (15):

$$\begin{aligned} \sum_{i=1}^n D_{active} &= D_{active}(1) + D_{active}(2) + \dots + D_{active}(n) \\ &= k_1 + 2 * (\alpha + \beta) + k_2 \dots + (n - 1) * 2 * (\alpha + \beta) + k_3 \end{aligned} \quad (15)$$

### 3.2.2 Analytical model for Data collection with aggregation

Like *section 3.2.1*, an estimation model is required for data collection with aggregation. The computation for the durations of activity and sleep are presented hereafter, as well as the energy consumption of estimation model for nodes and the whole network.

In this approach, we also begin with the furthest node. It first sends messages to parent node after an IFS ( $\beta$ ), and then waits for an *ACK* message. However, intermediate node (node 2) does not retransfer messages as in the approach without aggregation. Instead of, only an aggregated message is transmitted in order to reduce transmission cost. Therefore, an intermediate node first replies an *ACK* to its child with  $(\alpha + \beta)$  and then aggregates received message before the transmission. The details are given in the following paragraph. Once received the *ACK* message ( $\alpha$ ), the furthest node shifts its radio module to sleep mode and records the sleep time. Therefore, the total active duration of node 1 is  $2\alpha + 2\beta$ .

Activity duration at intermediate nodes is also in three parts for data collection with aggregation: First, it sends an *ACK* message to its child  $(\alpha + \beta + \alpha)$ . Second, it aggregates the received message from child node with its own message. Third, it transmits the aggregated message and waits for the reply from its parent  $(2\alpha + 2\beta)$ . Compared with the transmission time, data processing at CPU for the second time is small. Hence, we neglect it to build the model. At the end, the duration of node  $i$  for data collection in this approach can be expressed as in (16):

$$D_{active}(i) = \begin{cases} 2\alpha + 2\beta, & \text{when } i = 1 \\ 4\alpha + 3\beta + d_{syn}, & \text{when } i \neq 1 \end{cases} \quad (16)$$

Similar to the approach without aggregation, each node will set its communication module into sleep mode after receiving an *ACK* message from its parent. Therefore, the duration of sleep for node  $i$  can be computed with (11). The constant  $4\alpha + 3\beta + \gamma$  can be replaced by  $k_2$ . The total active duration can be calculated in (17):

$$\begin{aligned} \sum_{i=1}^n D_{active}(i) &= D_{active}(1) + D_{active}(2) + \dots + D_{active}(n) \\ &= 2\alpha + 2\beta + 4\alpha + 3\beta + d_{syn} + \dots + 4\alpha + 3\beta + d_{syn} \\ &= 2\alpha + 2\beta + (n-1) * (3\alpha + 4\beta + d_{syn}) \\ &= 2\alpha + 2\beta + (n-1) * k_2 \end{aligned} \quad (17)$$

The energy consumption of the whole network can be computed as in (18):

$$E_{total} = n * T * P_{sleep} + \sum_{i=1}^n D_{active}(i) * (P_{active} - P_{sleep}) \quad (18)$$

$$= n * T * P_{sleep} + (2\alpha + 2\beta + (n - 1) * k_2) * (P_{active} - P_{sleep})$$

With mentioned models, we can then estimate energy consumption for data collection with or without aggregation in a chain structure. The size of chain ( $N$ ), the powers of radio module ( $P_{active}, P_{sleep}$ ) and data collection period ( $T$ ) depend on used communication module and the data collection scenarios. Due to the difference of active duration, energy consumption of nodes is different. A comparative estimation result for a 20-node chain is shown in Figure 48 where data collections starts from the node 1 until the node 20. The parameters  $\alpha, \beta, d_{syn}$  are fixed at 20 ms, 100 ms, and 10 ms respectively. The closer to the sink node, the longer its active duration for data collection without aggregation. Unlike the linear growth in active duration for the case without aggregation, all intermediate nodes have the same active duration for data collection with aggregation. In the case of data collection without aggregation, the more nodes there are, the more obvious the hot-spot phenomenon is. Therefore, using data aggregation techniques at intermediate nodes may be useful for extending network lifetime. This analytical energy model is compared with a physical platform to verify its accuracy (tests and results are described in section 5.1).

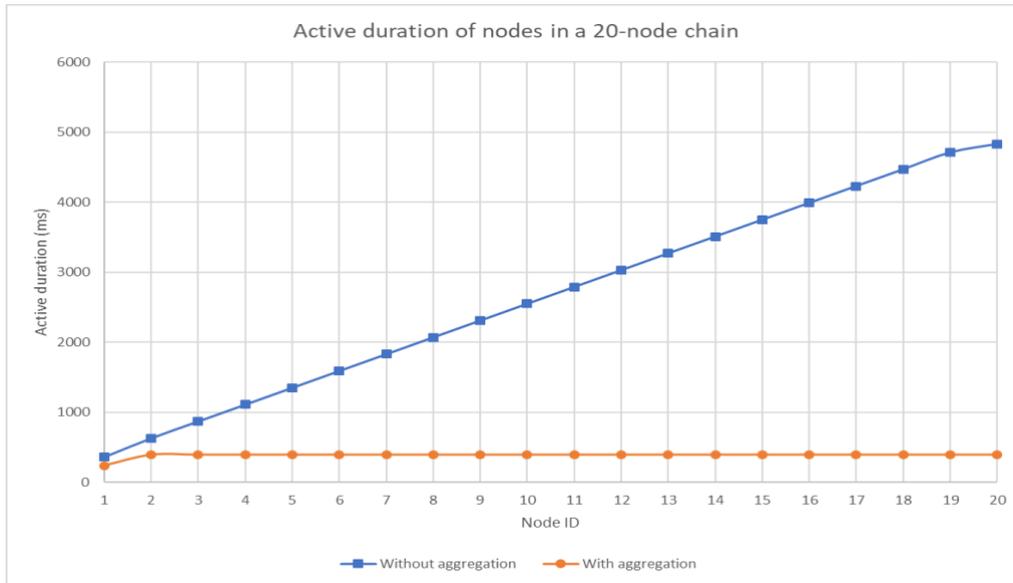


Figure 48 Estimation of active duration for chain-based approach

### 3.3 Analytical model for data aggregation in WSN

#### 3.3.1 Energy consumption estimation method in WSN

The previous study shows that using aggregation at intermediate nodes could reduce energy consumption as mentioned above. As consequence, aggregation methods (lossless and lossy)

are considered for building a generic analytical energy estimation model which can be applied for all WSN. Therefore, a new analytical model based on the communication relations between nodes and on the application of in-network processing at nodes is proposed hereafter.

The framework for the generic energy consumption model is presented in Figure 49. Some elements are introduced to compute energy consumption of nodes:

- The *relationship matrix*  $P$  records the communication relationship between nodes which depends on the used data collection routing protocol;
- The *aggregation coefficient*  $\lambda$  depends on the selected aggregation method by users;
- The *gathering matrix*  $G$  stores aggregation processing method and the gathering path, it can be calculated with  $P$  and with aggregation coefficients  $\lambda$ ;
- $M_{rx}$  and  $M_{tx}$  are vectors where each element represents respectively the *size of received message* and *transmitted message* for each node. They can be computed by using the matrices  $P$  and  $G$ .
- *Active duration*  $D$  is a vector recording the total duration in active state for each node, it can be computed with  $M_{rx}$  and  $M_{tx}$ .
- *Energy consumption*  $E$  is a vector recording the estimated energy consumption of nodes per round, which depends on the active duration and the communication module.

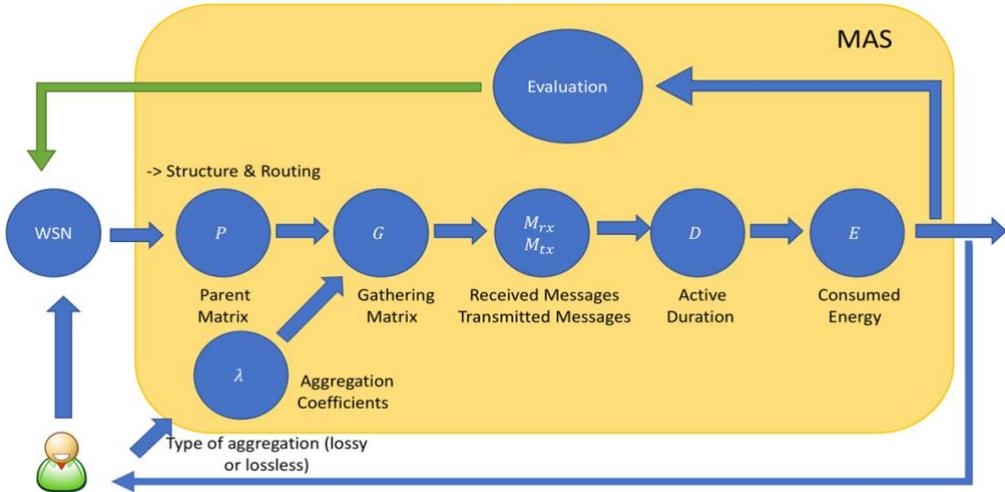


Figure 49 WSN energy consumption method's overview

To compute active duration for a node  $i$ , it is necessary to first calculate the size of received message  $M_{rx}(i)$  and transmitted message  $M_{tx}(i)$  at nodes. The transmitted message depends on the received messages, the monitoring data at nodes and the aggregation method. We assume that the format and the size of monitoring data are the same at each node. Each node can process

its received messages and its own message with same or different aggregation coefficients as in described Figure 50. For example, the intermediate node  $i$  has two children. The size of transmitted message from node  $j$  and node  $k$  are  $S_1$  and  $S_2$ , and the size of monitoring data at the node  $i$  is  $S_3$ . Therefore, the size of received message  $M_{rx}(i)$  at node  $i$  is the sum of the messages from  $j$  and  $k$  ( $S_1$  and  $S_2$ ).

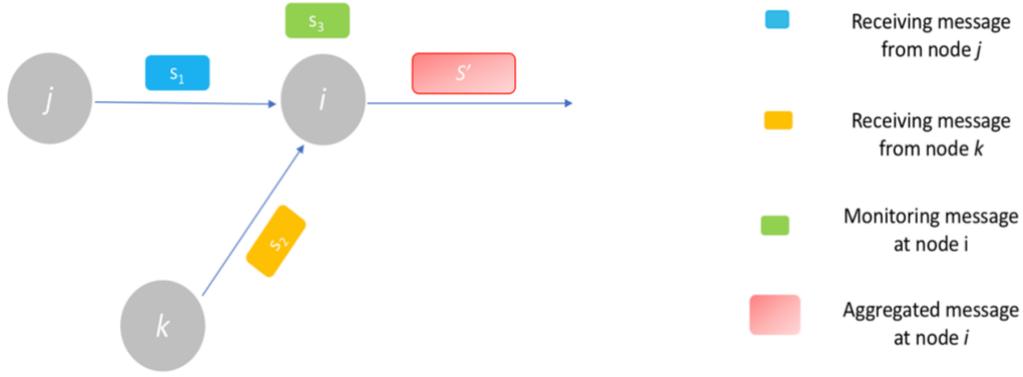


Figure 50 Aggregation coefficients at node

Node  $i$  can aggregate received message and its own monitoring data with different aggregation coefficients  $\lambda_{ij}$ ,  $\lambda_{ik}$ ,  $\lambda_{ii}$ . **The aggregation coefficient  $\lambda_{ij}$  represents the aggregation method for processing the message from the node  $j$  towards node  $i$ .**

In case of **lossless data aggregation**, nodes compress or merge all received messages and its own monitoring data in one or multiple messages (depending on the maximum length of packet size), all raw data is retained and transferred until the sink node. Therefore, **all aggregation coefficients for a node  $i$  are equal to 1 as in (19):**

$$\lambda_{ij} = 1, j \in \{\text{children of the node } i\} \quad (19)$$

In contrast, all the original data values cannot be recovered in **lossy data aggregation** (like the aggregation functions: mean, max or min). Each node sends only aggregated and data reduced messages. In case of a node  $i$  has  $m$  children, it can merge received messages and its own message **with the same aggregation coefficient  $1/(m + 1)$**  into an aggregated message. **The sum of these aggregation coefficients is equal to one.** Therefore, aggregation coefficients at node  $i$  can be expressed as in (20):

$$\lambda_{ii} = \lambda_{ij} = 1/(m + 1), j \in \{\text{children of the node } i\} \quad (20)$$

$$\lambda_{ii} + \sum_{j \in \{\text{children of the node } i\}} \lambda_{ij} = 1$$

An example for lossy aggregation is shown in Figure 51 where the node  $i$  has two children, the node  $j$  and the node  $k$ . All nodes report their temperature to the next node and only the maximal temperature is reported until the sink. In this example, the node  $i$  receives the sampling information from its children (15 degrees for node  $j$  and 16 degrees for node  $k$ ), it then compares the received information and its own monitoring information (15.5 degrees). Only the maximal temperature is reported to next node, therefore, all aggregation coefficients at node  $i$  are  $1/3$ .

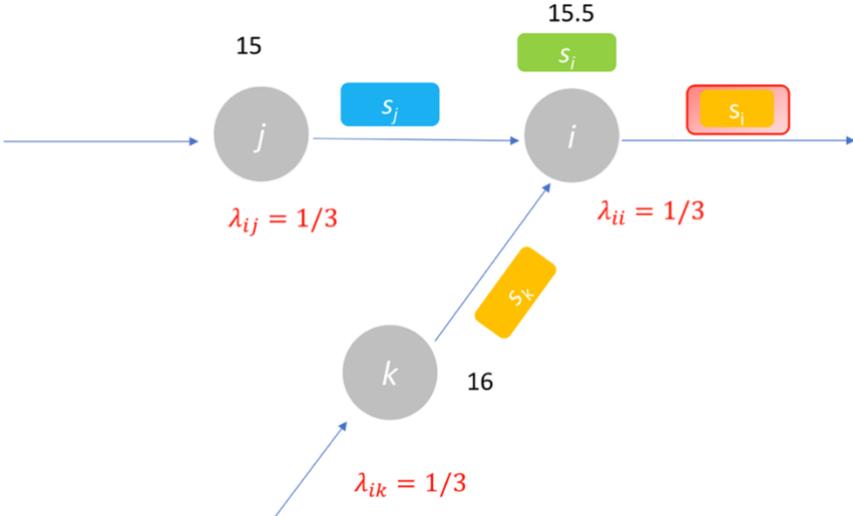


Figure 51 Data collection example with lossy aggregation

The size of a data packet may vary from minimal size ( $S_{min}$ ) to maximum size ( $S_{max}$ ). A data packet includes at least the header part and the data part of node, where the header part records the node ID, destination ID and other communication information. The data part could record the monitoring data of a node, request or reply information like *ACK* and so on. The sizes of  $S_{min}$  and  $S_{max}$  depend on the applied communication technologies. One more message is transmitted if the data part of a packet is full (when transmitted data is larger than  $S_{max}$ ). Corresponding transmitting durations are illustrated in Figure 52. From the minimal size to maximum size, the transmitting duration increases linearly from the minimal transmitting duration  $\alpha_{min}$  to the maximum transmitting duration  $\alpha_{max}$ . To estimate the total active duration, the size of transmitted and received messages are detailed in the following.

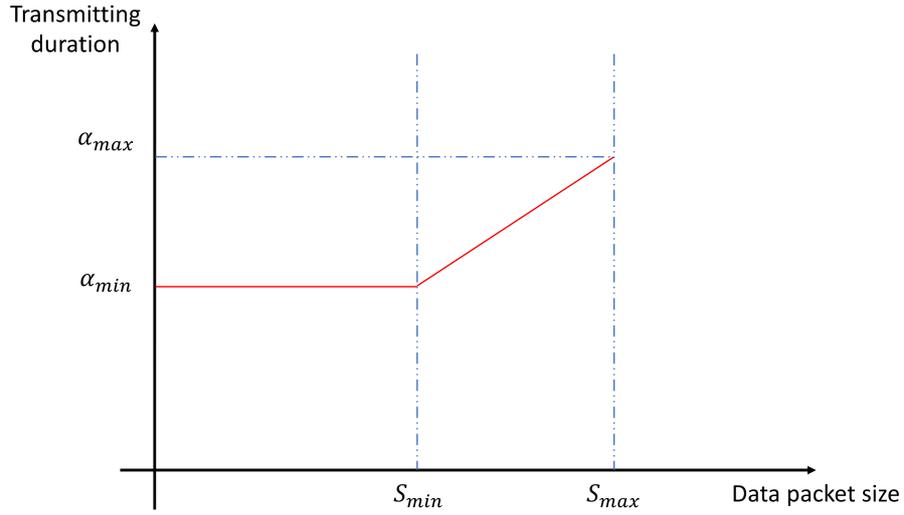


Figure 52 Transmitting duration for different size of data packet

The size of transmitting message depends on the gathering path and on the applied aggregation method. To generalize the relationship between nodes in the network, parent relationship matrix  $P$  is used to record the interaction between child nodes and their parent as in (21). For an intermediate node  $i$ ,  $P_{ij}$  is equal to  $I$  when the node  $i$  is the parent node of the node  $j$ .

$$P = \begin{bmatrix} P_{11} & \cdots & P_{1j} & \cdots & P_{1N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{i1} & \cdots & P_{ij} & \cdots & P_{iN} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{N1} & \cdots & P_{Nj} & \cdots & P_{NN} \end{bmatrix} \quad (21)$$

Based on the relationship matrix between nodes and on the aggregation coefficients, a gathering matrix  $G$  is used to record the collection method as in (22). The element  $G_{ij}$  records message processing coefficient from the node  $j$  at the node  $i$ .

$$G = \begin{bmatrix} G_{11} & \cdots & G_{1j} & \cdots & G_{1N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ G_{i1} & \cdots & G_{ij} & \cdots & G_{iN} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ G_{N1} & \cdots & G_{Nj} & \cdots & G_{NN} \end{bmatrix} \quad (22)$$

For example, in a chain structure as shown in Figure 53,  $P_{j,j+1} = I$ . We suppose that the size of the sent message from node  $j$  is  $k$  and the data aggregation coefficient at a node  $j+1$  is  $\lambda_{j+1j}$ . Therefore, the processing coefficient  $G_{ij}$  in this chain structure can be expressed as in (23).



Figure 53 Data processing flow of the message from node  $j$

$$G_{ij} = \lambda_{j+1,j} * \lambda_{j+2,j+1} * \dots * \lambda_{i,i-1} = \prod_j^{i-1} \lambda_{j+1,j} \quad (23)$$

With the equation (23), the computation of the gathering matrix can be generalized as in **Algorithm 1** where the relationship  $P$  and the data aggregation coefficients of all nodes are used to automatically generate the gathering matrix  $G$ . The algorithm is shown as follows:

**Algorithm 1. Gathering Matrix Generation algorithm**

---

```

1:   E is the vector of end nodes;
2:   K is the ID of one end node from E;
3:   H is the ID of leader;
4:   N is the total number of nodes
5:   i = K;
6:   Temp;
7:
8:   For (j = 1; j ≤ N; j++) {
9:       G[j][j] = λjj; } //Data aggregation coefficient for its own message
10:
11:  While ( i ≠ H){
12:      For (j = 1; j ≤ N; j++) {
13:          If (P[j][i] == 1) then //Search its parent node
14:              Temp = j;
15:          For (m = 1; m ≤ N; m++) { //Add aggregation coefficient for children
16:              If (G[i][m] != 0) then
17:                  G[j][m] = λji * G[i][m];
18:          }
19:          i = temp;
20:      }
21:      Remove K from E;
22:      Repeat until E is null;

```

---

The received message at a node is the sum of the aggregated messages from their child nodes. And its transmitted message can be computed by the sum of received messages and its own monitoring message with their corresponding aggregation coefficients as in (24), where vectors  $M_{tx[N]}$  and  $S_{[N]}$  record the size of transmitted message and monitoring data size at nodes. Based on the vector  $M_{tx[N]}$  and the relationship matrix  $P$ , the size of received message for a node  $i$  can then be computed as in (25).

$$M_{tx} = G \times S = \begin{bmatrix} G_{11} & G_{12} & \dots & G_{1N} \\ \vdots & \vdots & \vdots & \vdots \\ G_{i1} & G_{i2} & \dots & G_{iN} \\ \vdots & \vdots & \vdots & \vdots \\ G_{N1} & G_{N2} & \dots & G_{NN} \end{bmatrix} \begin{bmatrix} S_1 \\ \vdots \\ S_2 \\ \vdots \\ S_N \end{bmatrix} \quad (24)$$

$$= \begin{bmatrix} G_{11}S_1 + \dots + G_{ii}S_i + \dots + G_{1N}S_N \\ \vdots \\ G_{i1}\lambda_{11} + \dots + G_{ii}\lambda_{ii} + \dots + G_{iN}\lambda_{NN} \\ \vdots \\ G_{N1}\lambda_{NN} + \dots + G_{ii}\lambda_{ii} + \dots + G_{NN}\lambda_{NN} \end{bmatrix}$$

$$M_{rx} = P \times M_{tx} \quad (25)$$

For example, the gathering matrix for a 3-node chain where node 1 transmits data to node 2 until the BS (node 3) can be computed as in (26). With this matrix and the monitoring data size vector  $S_{[N]}$ , the vector of transmitted message can be computed as in (27). Using the transmitted message matrix and the relationship matrix, the received message matrix can be computed as in (28).

$$G = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{bmatrix} = \begin{bmatrix} \lambda_{11} & 0 & 0 \\ \lambda_{21} & \lambda_{22} & 0 \\ \lambda_{21}\lambda_{32} & \lambda_{32} & \lambda_{33} \end{bmatrix} \quad (26)$$

$$M_{tx} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} \quad (27)$$

$$= \begin{bmatrix} \lambda_{11} & 0 & 0 \\ \lambda_{21}\lambda_{11} & \lambda_{22} & 0 \\ \lambda_{32}\lambda_{21}\lambda_{11} & \lambda_{32}\lambda_{22} & \lambda_{33} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix}$$

$$= \begin{bmatrix} S_1\lambda_{11} + 0 + 0 \\ \lambda_{21}\lambda_{11}S_1 + \lambda_{22}S_2 + 0 \\ \lambda_{32}\lambda_{21}\lambda_{11}S_1 + \lambda_{32}\lambda_{22}S_2 + \lambda_{33}S_3 \end{bmatrix}$$

$$M_{rx} = P \times M_{tx} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} S_1\lambda_{11} \\ \lambda_{21}\lambda_{11}S_1 + \lambda_{22}S_2 \\ \lambda_{32}\lambda_{21}\lambda_{11}S_1 + \lambda_{32}\lambda_{22}S_2 + \lambda_{33}S_3 \end{bmatrix} \quad (28)$$

$$= \begin{bmatrix} 0 \\ S_1\lambda_{11} \\ \lambda_{21}\lambda_{11}S_1 + \lambda_{22}S_2 \end{bmatrix}$$

For a given network, the gathering matrix  $G$  can be computed by the network relationship between nodes (the matrix  $P$ ) and the data aggregation coefficients. This gathering matrix can be used to calculate received and transmitted messages sizes. The total active duration of a node  $i$  can be computed in (29). The transmitting duration  $D_{tx}$  depends on the size of transmitted data packet. Messages are merged as much as possible into aggregated messages according to the maximal length. The receiving duration  $D_{rx}$  depends on the size of received data packet and

the number of children, because an *ACK* message is replied to each child after the IFS ( $\beta + \gamma$ ). With the total active duration, energy consumption of this node can then be computed as follows:

$$D_{active}(i) = D_{rx}(i) + \beta + D_{tx}(i) \text{ (see Figure 47)} \quad (29)$$

$$\text{Where, } D_{tx}(i) = \begin{cases} \left\lfloor \frac{M_{tx}(i)}{S_{max}} \right\rfloor * \alpha_{max}, & \text{if } \left\lfloor \frac{M_{tx}(i)}{S_{max}} \right\rfloor = 0 \\ \left\lfloor \frac{M_{tx}(i)}{S_{max}} \right\rfloor * \alpha_{max} + \max\left(\alpha_{min}, \left\lfloor \frac{M_{tx}(i) \bmod S_{max}}{S_{min}} \right\rfloor * \alpha_{min}\right), & \text{if } \left\lfloor \frac{M_{tx}(i)}{S_{max}} \right\rfloor \neq 0 \end{cases}$$

$$D_{rx}(i) = \sum_{\forall j \in \{\text{children of } i\}} D_{tx}(j) + \beta + \gamma$$

$$\begin{aligned} E_i &= D_{active}(i) * P_{active} + D_{sleep}(i) * P_{sleep} \\ &= D_{active}(i) * P_{active} + (T - D_{active}(i)) * P_{sleep} \\ &= P_{sleep} * T + D_{active}(i) * (P_{active} - P_{sleep}) \\ &= P_{sleep} * T + (D_{rx}(i) + \beta + D_{tx}(i)) * (P_{active} - P_{sleep}) \end{aligned}$$

### 3.3.2 Model for tree structure

In some phases of CM lifetime, different communicating concretes may cooperate with each other to form a tree structure. The example of a tree structure with  $N$  nodes is shown as Figure 54 where the node  $i$  is the root node. All nodes send their message to their parent until the root node is reached.



Figure 54 Root node is an intermediate node during data collection

Data collection in this type of network can be considered as the collection of two small chains, an example of 5-node tree structure is shown Figure 55. These two small chains can perform data collection simultaneously, e.g., by using TDMA as in LEACH (Heinzelman et al., 2002) or step by step with a token in PEGASIS (Lindsey et al., 2002). In our study, we focus on data collection with TDMA methods. Before the phase of data collection, nodes are synchronized. All nodes can turn on the communication module at the scheduled time to receive and transmit messages. The chronogram example for this 5-node tree structure is illustrated in Figure 56.

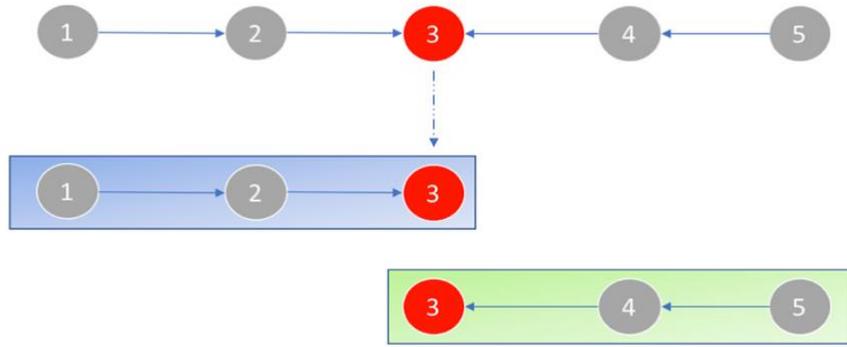


Figure 55 Gathering two chains into a tree structure

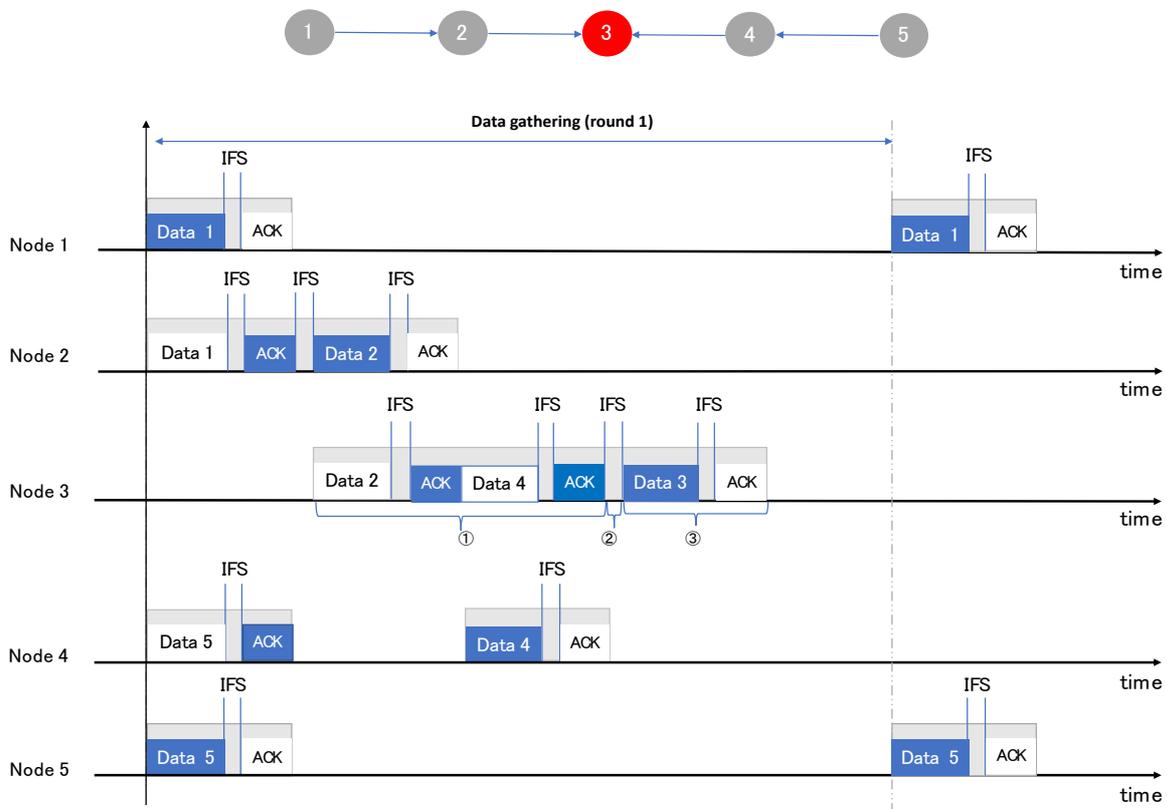


Figure 56 Periodical data collection in tree structure

Nodes in this tree structure have the same activities than two small chains, except for the root node. The active duration of root node (the node 3) is composed of three different parts: the reception of the messages sent by its children (identified as ① in the chronogram); the transmission of aggregated messages (identified as ③ in the chronogram); and an IFS between both (identified as ② in the chronogram).

Concerning the reception, the node 3 has two child nodes. The active duration for receiving and replying *ACK* messages ( $\gamma$ ) is thus double than for a single chain as in (30). As a

consequence, the duration for reception of node  $i$  with many children can be computed as in (30) where  $D_{tx}(j)$  is the transmitting duration of one of its children. To ensure that nodes can wake up at the right time, a synchronization delay is introduced in (31).

$$D_{rx}(i) = \sum_{j \in \{\text{children of } i\}} (D_{tx}(j) + \beta + \gamma) \quad (30)$$

$$D_{rx}(i) = d_{syn} + \sum_{j \in \{\text{children of } i\}} (D_{tx}(j) + \beta + \gamma) \quad (31)$$

Unlike the single chain structure, the gathering matrix in tree (or multi-chain) structure is more complex. For the example of the Figure 57, the node 3 can receive messages from the node 2 and node 4. The gathering matrix  $G$  of the 5-node tree can be seen as the fusion of the two small chains via the node 3, where  $G_1$  and  $G_2$  are the gathering matrix for the chain from node 1 to node 3, and node 5 to node 3 respectively.

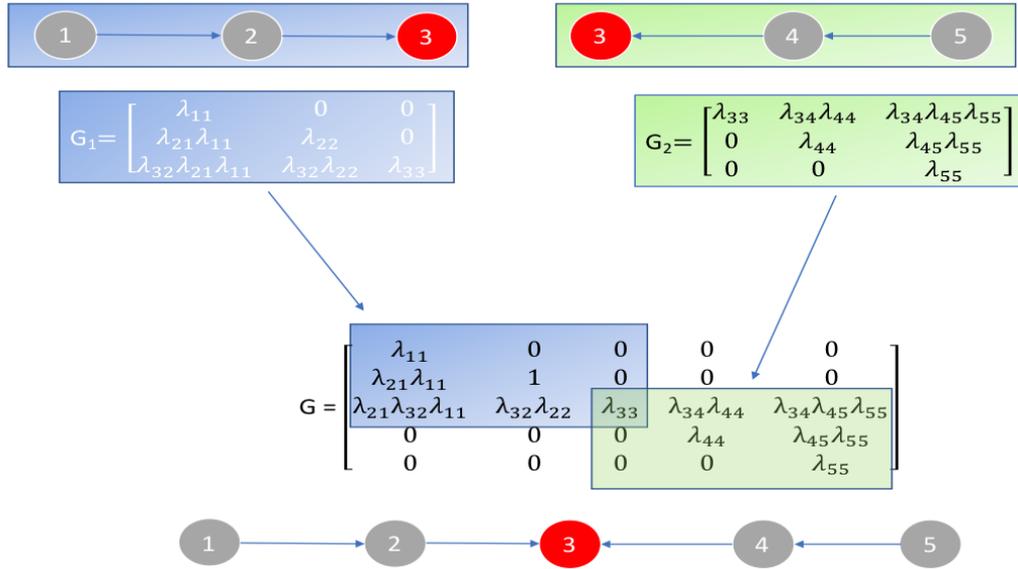


Figure 57 Tree as a composition of chains

With the gathering matrix as in (32), the transmitted matrix can then be computed as in (33), and the received message matrix can be obtained with the equation in (28). As consequence, active duration for all nodes can then be calculated with the equation in (29).

$$G = \begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} & G_{15} \\ G_{21} & G_{22} & G_{23} & G_{24} & G_{25} \\ G_{31} & G_{32} & G_{33} & G_{34} & G_{35} \\ G_{41} & G_{42} & G_{43} & G_{44} & G_{45} \\ G_{51} & G_{52} & G_{53} & G_{45} & G_{55} \end{bmatrix} = \begin{bmatrix} \lambda_{11} & 0 & 0 & 0 & 0 \\ \lambda_{21}\lambda_{11} & \lambda_{22} & 0 & 0 & 0 \\ \lambda_{32}\lambda_{21}\lambda_{11} & \lambda_{32}\lambda_{22} & \lambda_{33} & \lambda_{34}\lambda_{44} & \lambda_{34}\lambda_{45}\lambda_{55} \\ 0 & 0 & 0 & \lambda_{44} & \lambda_{45}\lambda_{55} \\ 0 & 0 & 0 & 0 & \lambda_{55} \end{bmatrix} \quad (32)$$

$$\begin{aligned}
M_{tx} &= \begin{bmatrix} \lambda_{11} & 0 & 0 & 0 & 0 \\ \lambda_{21}\lambda_{11} & \lambda_{22} & 0 & 0 & 0 \\ \lambda_{32}\lambda_{21}\lambda_{11} & \lambda_{32}\lambda_{22} & \lambda_{33} & \lambda_{34}\lambda_{44} & \lambda_{34}\lambda_{45}\lambda_{55} \\ 0 & 0 & 0 & \lambda_{44} & \lambda_{45}\lambda_{55} \\ 0 & 0 & 0 & 0 & \lambda_{55} \end{bmatrix} \times \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{bmatrix} \\
&= \begin{bmatrix} \lambda_{11}S_1 \\ \lambda_{11}\lambda_{21}S_1 + \lambda_{22}S_2 \\ \lambda_{11}\lambda_{21}\lambda_{32}S_1 + \lambda_{21}\lambda_{32}S_2 + \lambda_{33}S_3 + \lambda_{34}\lambda_{45}S_4 + \lambda_{34}\lambda_{45}\lambda_{55}S_5 \\ \lambda_{45}\lambda_{55}S_5 + \lambda_{44}S_4 \\ \lambda_{55}S_5 \end{bmatrix}
\end{aligned} \tag{33}$$

For a  $N$ -node tree where the node  $i$  is the root node, we can then generate the gathering matrix as in (34) which includes the matrices of its two parts.

$$G = \begin{bmatrix} \lambda_{11} & \dots & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \lambda_{11} * \dots * \lambda_{i-1i-2} & \dots & \ddots & 0 & 0 & 0 \\ \lambda_{11} * \lambda_{21} * \dots * \lambda_{ii-1} & \dots & \lambda_{ii}\lambda_{ii-1} & \lambda_{ii} & \dots & \lambda_{ii+1} * \dots * \lambda_{N-1N}\lambda_{NN} \\ 0 & \dots & 0 & 0 & \ddots & \vdots \\ \vdots & \dots & 0 & 0 & 0 & \lambda_{N-1N}\lambda_{NN} \\ 0 & \dots & 0 & 0 & 0 & \lambda_{NN} \end{bmatrix} \tag{34}$$

### 3.3.3 Model for other structures

Independent chain networks can be grouped to form a tree structure or a cluster structure. An example is shown in Figure 58. There are two CM elements: a data collection tree structure is used as in element 1 (the red block) or a cluster structure in element 2 (the blue block). In a case of cooperation between the two elements, the node 9 which is the cluster leader links the two parts via the node 4 (the link in orange). The node 4 then transmits data to the BS via the node 9 instead of transmitting directly to the BS. Thus, nodes in each CM element send their information to the next until the cluster leader via their own topology structure.

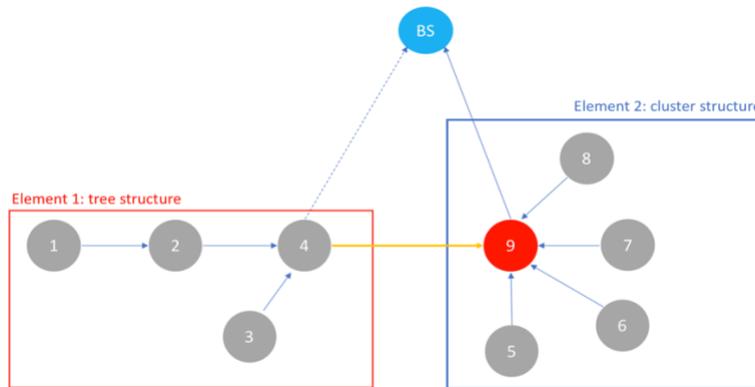


Figure 58 Data collection in different structures

With the **Algorithm 1**, the matrix for the element 1 in red and element 2 in blue can be generated as in (35) and (36) respectively. The gathering matrix for this network can be seen as the composition of the two parts as illustrated in (37), all messages from the node 4 are aggregated at the node 9 with the aggregation coefficient  $\lambda_{94}$  (in orange color). Therefore, the gathering element  $G_{91}$  of the new gathering matrix in (37) can be computed with the gathering element  $G_{41}$  of  $G_1$  in (35) and the responding aggregation coefficient  $\lambda_{94}$ , same for other messages from the nodes of element 1. We can conclude that gathering matrix for a composed network can be seen as the composition of the independent small matrices. In this example, the node 9 links the two structures without changing the structures. In other cases, there may be modifications in the communication structure and in the routing of data.

$$G_1 = \begin{bmatrix} \lambda_{11} & 0 & 0 & 0 \\ \lambda_{21}\lambda_{11} & \lambda_{22} & 0 & 0 \\ 0 & 0 & \lambda_{33} & 0 \\ \lambda_{42}\lambda_{21}\lambda_{11} & \lambda_{42}\lambda_{22} & \lambda_{43}\lambda_{33} & \lambda_{44} \end{bmatrix} \quad (35)$$

$$G_2 = \begin{bmatrix} \lambda_{55} & 0 & 0 & 0 & 0 \\ 0 & \lambda_{66} & 0 & 0 & 0 \\ 0 & 0 & \lambda_{77} & 0 & 0 \\ 0 & 0 & 0 & \lambda_{88} & 0 \\ \lambda_{95}\lambda_{55} & \lambda_{96}\lambda_{66} & \lambda_{97}\lambda_{77} & \lambda_{98}\lambda_{88} & \lambda_{99} \end{bmatrix} \quad (36)$$

$$G = \begin{bmatrix} \lambda_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_{21}\lambda_{11} & \lambda_{22} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_{33} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_{42}\lambda_{21}\lambda_{11} & \lambda_{42}\lambda_{22} & \lambda_{43}\lambda_{33} & \lambda_{44} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{55} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_{66} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{77} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{88} & 0 & 0 \\ \lambda_{94}\lambda_{42}\lambda_{21}\lambda_{11} & \lambda_{94}\lambda_{42}\lambda_{22} & \lambda_{94}\lambda_{43}\lambda_{33} & \lambda_{94}\lambda_{44} & \lambda_{95}\lambda_{55} & \lambda_{96}\lambda_{66} & \lambda_{97}\lambda_{77} & \lambda_{98}\lambda_{88} & \lambda_{99} & 0 \end{bmatrix} \quad (37)$$

Based on gathering matrix  $G$ , we can then apply the equations (24) and (25) to calculate the size of transmitted and received messages. Messages sizes can then be used to compute active durations in order to estimate their consumed and residual energy with the equation (29). Lifetime computation examples are given in *Chapter 4* and *Chapter 5*.

### 3.4 Conclusion

In this chapter, a simple energy model for communicating node is presented, followed by the explanation of the data collection process in chain-based structure. The energy consumption models for data collection with/without aggregation in a chain structure are detailed. The

accuracy of this position-based analytical model in chain is verified by a physical platform. The experimental results are presented in *Chapter 5 (section 5.1)*. In our proposal, the energy spent by the radio interface of a node only depends on five parameters (consumption is uniform all along the chain in case of collection with aggregation) and also depends on the position of the node in the chain (in case of collection without aggregation). The general formulation of the energy consumed by a node in a chain is defined as follows:

$$E_{consumed} = f(\alpha, \beta, \gamma, P_s, P_a)$$

With:

- $P_s$ , the power consumed by radio in sleep mode;
- $P_a$ , the power consumed by radio in active mode;
- $(\alpha, \beta, \gamma)$ , the delay parameters, , expressing the active duration of radio module.

Figure 59 gives a visual representation of this model for an individual node transmitting data during one collection cycle.  $P_s$  gives the tendency between two collection periods.  $P_a$  gives the tendency during a collection cycle. The active duration of radio depends on  $(\alpha, \beta, \gamma)$ . The values of these parameters can be adapted for any wireless network communication technology. Active duration is fixed for data collection using aggregation. In case of a collection without aggregation, the active duration exponentially increases with the position of node to the sink node (revealing the well-known hot-spot problem in chain structure).

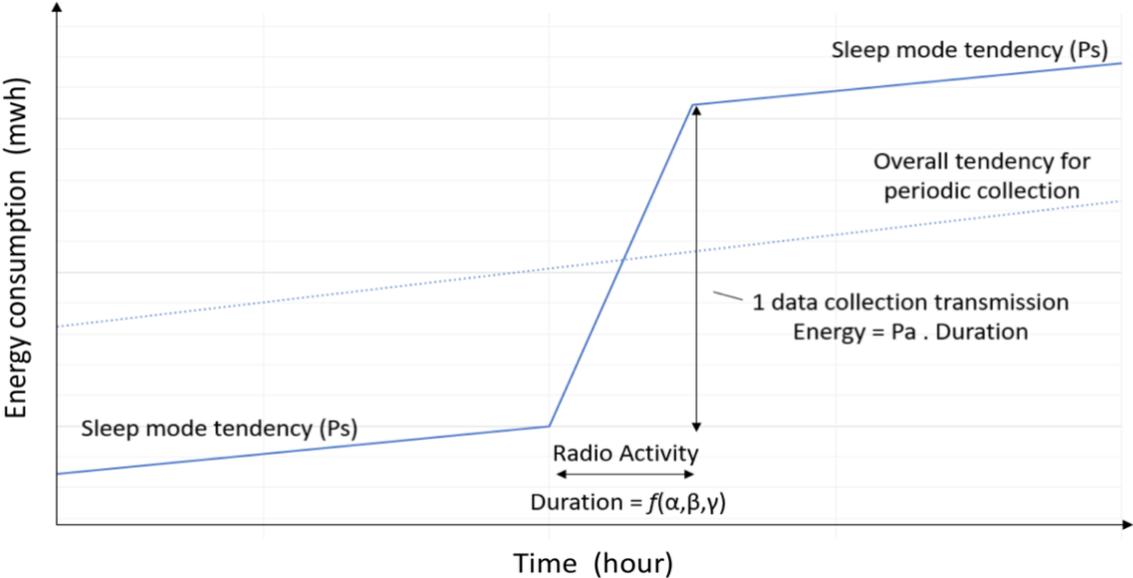


Figure 59 Radio interface consumption model for a node (one collection cycle)

If all nodes have the same material characteristics, tendencies (slopes induced by  $P_s$  and  $P_a$ ) should be the same for all nodes involved in a chain. The duration is the only variable changing and impacting the energy consumption of each node. Our models allow quantifying the potential energy gain involved by using (or not using) aggregation for chain data collection. Because these parameters do not evolve in time and if, during application, the required collection frequency does not change too, the total energy spent by a node can be directly obtained with the number of collection cycle. In this approach, the consumed energy at node depends on device characteristics ( $P_a$  and  $P_s$ ), on the communication structure (chain) and on the collection and network communication protocols (activity duration).

In addition, a relationship based analytical consumption model is proposed for all types of WSN structures. The relationship matrix  $P$  depends on the used routing or network structuring protocol. The gathering matrix  $G$  can be calculated with  $P$  and with aggregation coefficients  $\lambda$  at nodes. The size of received message  $M_{rx}$  and transmitted message  $M_{tx}$  can be computed by the matrix  $P$  and  $G$ . Messages sizes are used to calculate duration in active state to estimate residual energy for each node.

In conclusion, these simple and effective models can be used online by nodes for remaining energy self-assessment. Distributed routing algorithms can use the prediction consumption to determine or reconfigure the communication structure. Models can also be used by an external software control system to have a global and centralized vision of the spent energy of all nodes of the network. The cooperation between elements is analyzed via a multi-agent simulator (see section 5.2). To understand how the simulator was developed, next chapter (*Chapter 4*) presents and describes our second proposal: a recursive multi-agent architecture, and how it can be used to compose or decompose communicating materials along their lifecycles and/or to estimate residual energy of WSN nodes.



# Chapter 4

## Recursive architecture for energy management of Communicating Material

### Contents

<a href="#">4.1 The proposed CM recursive architecture</a> .....	87
<a href="#">4.2 Adapted MAS-R agent model</a> .....	89
<a href="#">4.3 Composition / decomposition mechanisms</a> .....	93
<a href="#">4.4 Interaction between agents for lifetime estimation</a> .....	96
<a href="#">4.5 Cooperation between two aggregated elements</a> .....	97
<a href="#">4.6 Energy consumption estimation scenarios</a> .....	100
<a href="#">4.6.1 Data collection with lossy data aggregation</a> .....	101
<a href="#">4.6.2 Energy consumption estimation for the composition of elements</a> .....	104
<a href="#">4.7 Conclusion</a> .....	107

Due to the limit of computing and storage at nodes, long-term service requirements bring challenges to the design of WSN in a resource constrained environment. Using a multi-agent system to build a digital twin can provide an optimal solution with a global view to extend network lifetime. This chapter begins with the recursive architecture for communicating material in *section 4.1*, followed by the proposed agent model in *section 4.2*. The composition/decomposition mechanisms are detailed in *section 4.3*. Interaction between agents for lifetime estimation is given in *section 4.4*. In some phases of the lifecycle, communicating concretes may cooperate with each other, different cooperation methods are then discussed in *section 4.5*. Besides, the proposed energy estimation model (proposed in *chapter 3*) is used for energy consumption estimation of a single concrete and an aggregated concrete in *section 4.6*. A conclusion is given in *section 4.7*.

### 4.1 The proposed CM recursive architecture

As introduced in *chapter 1*, the real (physical) node coupled with its agent (digital) node is an *elementary material holon* as shown in Figure 60. Due to the special composition ability of CM, these elementary holons can be grouped as a *composed material holon* via the *link 3*. The node

agents can communicate with the composed agent via the *link 1*. Besides, agents can discuss with those on the same level by exchanging messages via the interaction *link 2*. To organize the communication between agents on the same level or different levels, we would like to apply the MAS-R recursive architecture to our work as mentioned in chapter 2.

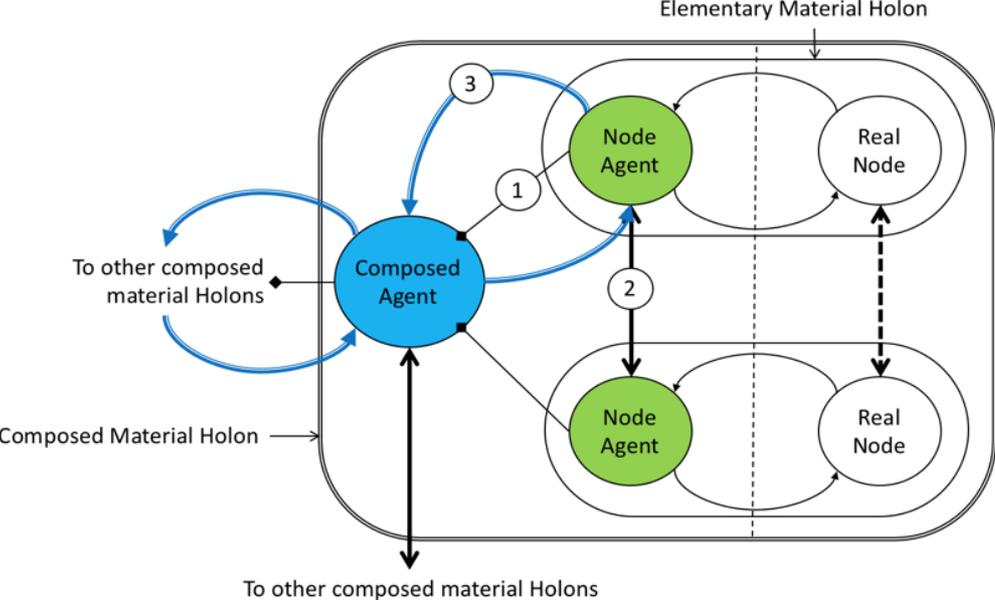


Figure 60 Composed CM agent

The CM recursive architecture for the two communicating elements example (tree structure in element 1 and cluster structure in element 2) in previous chapter is shown in Figure 61. There are four CNs (node 1 to 4) in the first element, and five CNs in the second (node 5 to 9). Each node is represented by an elementary agent (level 0). For an agent  $CM_i^j$ ,  $i$  and  $j$  represent the abstraction level and the index of the CM in the corresponding level respectively. Elementary agents (from  $CM_0^1$  to  $CM_0^9$ ) are aggregated into two composed CM agents ( $CM_1^1$  and  $CM_1^2$ ) on level 1. In the case of aggregation, those two level 1 composed agents are gathered into the agent  $CM_2^1$  at the level 2. The latter (and hierarchically higher) agent has the global view of all CM agents.

These links for building aggregated agent and communication between agents in the same level and different levels can be expressed by the operators  $PA$ ,  $PO$  and  $PI$  in MAS-R:

- The  $PA$  operator (Process Agent) is responsible for building aggregated agents or decomposing aggregated agents (the composition and decomposition processes will be detailed in section 4.3).

- The *PI* operator (Process Interaction) is defined for transforming interactions between different levels, for example, the interaction between agents  $CM_0^l$  and  $CM_1^l$  can be transformed into an interaction between agents  $CM_1^l$  and  $CM_2^l$  (an example will be given in *section 4.4*).
- the *PO* operator (Process Organization) is defined for transforming relations between agents of a level to the upper level (an example will be presented in *section 4.5*).

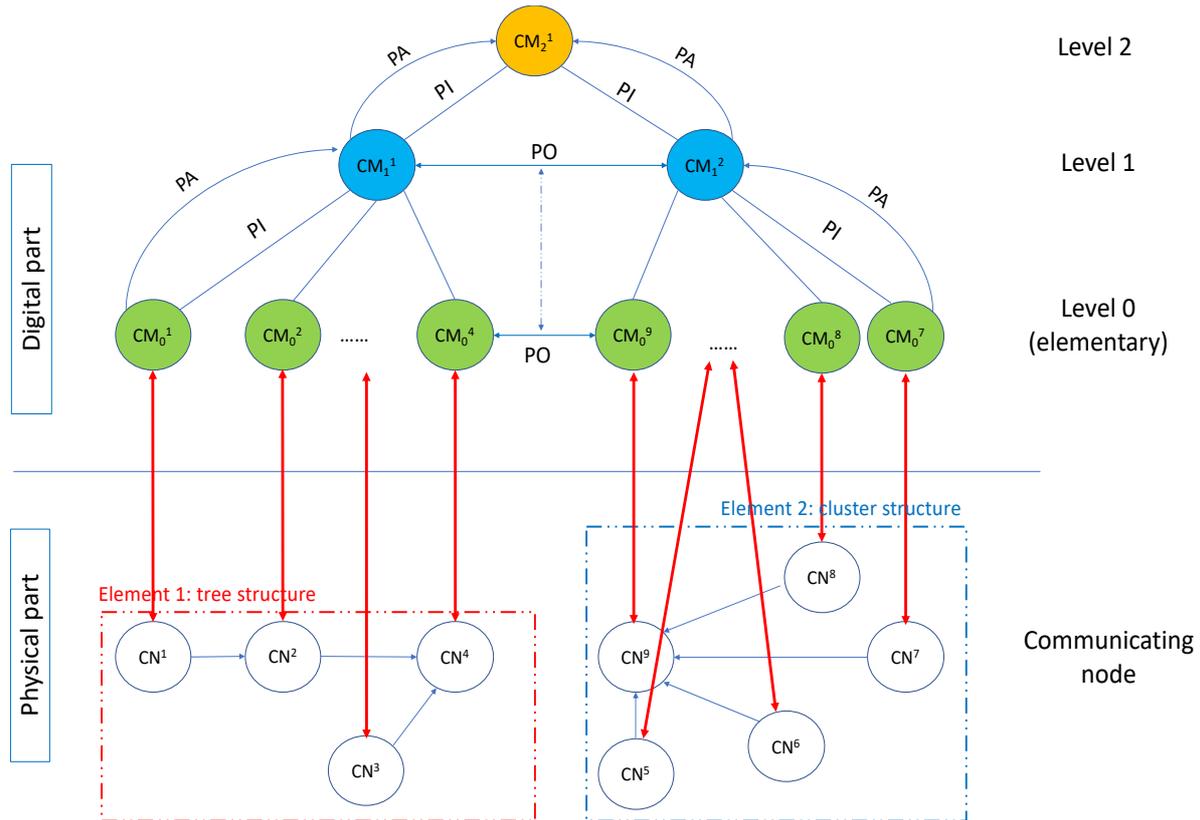


Figure 61 CM recursive architecture

## 4.2 Adapted MAS-R agent model

Based on the agent model in MAS-R model (Hoang, 2012), an adapted agent model, dedicated to the CM application, is proposed. The adapted CM agent still has five parts as shown in Figure 62: the knowledge part ( $K$ ) holding all data, the capabilities and recursive states of agent; the composition part ( $C$ ) and the decomposition part ( $D$ ) containing composition function to construct a higher level and decomposition function to destroy abstraction level; the observer part ( $O$ ) monitoring the recursion states of agent in  $K$  and call composition or decomposition functions if necessary; the recursive interaction part ( $RI$ ) recording the interaction functions for agents located at the same level and the interaction between different levels.

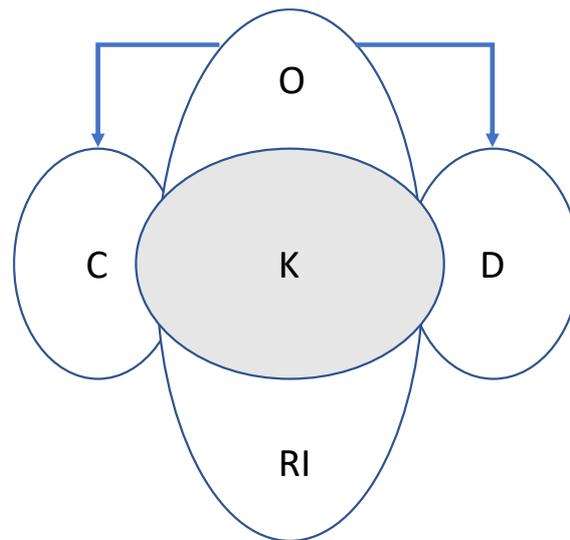


Figure 62 Agent model in MAS-R

Compared to the original agent model, the main difference is in the  $K$  part which includes two different types of data: product-related data and data for energy computation. In more details, the parameters of **product-related data** are presented as follows:

- *Node ID*: each communicating node have a unique identification in the embedded WSN of communicating concrete.
- *Current energy  $E_{res}$* : this information is different along the abstraction level, it includes remaining energy information of communicating node in level  $0$ . Otherwise, it includes energy information of all members in a vector.
- *Communicating element ID*: it is the identification of the communicating concrete. Each communicating concrete have a unique identification.
- *Agent ID*: it records the identification of an agent in the digital part where each communicating agent are represented by a node agent (elementary agent), the *ID* of an agent being unique in each abstraction level.
- *Abstraction level*: it is the agent level in the recursive architecture
- *Parent agent's ID*: recording the ID of composed agent which it belongs to.

To efficiently manage energy consumption of the embedded WSN, it is necessary to add all **energy-related information** to the  $K$  part of linked agents:

- *Communication relationship matrix  $P$*  (corresponding to the physical network): recording communicating relationship of a node in the data gathering structure (its children and parent nodes).

- *Cooperative agent*: the agents which can cooperate with, for example, the elementary node agent of the nodes in its communicating range. Those cooperative agents are recorded in the cooperative agent list (CAL).
- *Aggregation coefficients matrix  $\lambda$* : including the aggregation coefficients selected by the users.
- *Gathering matrix  $G$* : including the gathering matrix which depends on the communication relationship matrix  $P$  and the aggregation coefficients.
- *Transmitted matrix  $M_{tx}$  and received matrix  $M_{rx}$* : including the size of transmitted messages and received messages respectively.
- *Energy consumption per node  $E$* : it records the analytical energy consumption result of node in level  $0$ , or energy consumption of the whole network in upper levels.

In different phases of CM product lifecycle, aggregated agents can cooperate with others to complete special tasks or be decomposed into independent CM products. At the beginning of data collection, each physical node has a unique  $ID$  in the network, their physical information is stored in dedicated elementary agents. The knowledge (part  $K$ ) of agents on different abstraction levels are different. As an example, the knowledge of agents ( $CM_0^4$ ,  $CM_1^1$ ,  $CM_2^1$ ) for product-related information and energy consumption in three different levels is shown in Table V. a and Table V. b respectively. Only elementary agents contain the physical node  $ID$ . In this architecture, each agent has its unique identification  $CM_i^j$  which includes its abstract level  $i$  and its identification  $ID$   $j$  on this level. The communicating element  $ID$  is included for all agents. Besides, the agents in communication range are recorded in the  $CAL$ .

The relationship matrices  $P$  for agents located at different levels has different forms: relationship matrix of an elementary agent is a vector which includes its children  $ID$ . For example,  $P_0^4$  of the agent  $CM_0^4$  is shown in (38) where the node 4 is the parent for the node 2 and node 3. The higher the abstraction level, the higher the completeness of the relationship matrix as the agent  $CM_1^1$  has the global view of its members in (39). An agent in highest level has the global view of the whole network as the  $CM_2^1$  in (40). Based on relationship and aggregation coefficient matrices, an aggregated agent (agent with an abstract level greater than or equal to  $1$ ) can compute the gathering matrix  $G$  and energy consumption  $E$  and estimate network lifetime.

Table IV. a. Knowledge part  $K$  concerning product-related information

Agents	Node ID	Current Energy $E_{res}$	Element ID	Agent ID	Abstraction Level	Parent Agent
$CM_0^4$	4	[100%]	1	4	0	$CM_1^1$
$CM_0^5$	5	[100%]	2	5	0	$CM_1^2$
$CM_1^1$	0	$E_{res_1}^1$	1	1	1	$CM_2^1$
$CM_1^2$	0	$E_{res_1}^2$	2	2	1	$CM_2^1$
$CM_2^1$	0	$E_{res_2}^1$	1-2	1	2	0

Table V. b. Knowledge part  $K$  concerning energy consumption in different abstraction levels

Agents	Relationship Matrix $P$	CAL	Aggregation coefficient Matrix $\lambda$	Gathering Matrix $G_i$	Energy consumption $E_i$
$CM_0^4$	$P_0^4$	$CM_0^2, CM_0^3, CM_0^5$	$\lambda_0^4$	$G_0^4$	$E_0^4$
$CM_0^5$	$P_0^5$	$CM_0^6, CM_0^8, CM_0^9$	$\lambda_0^5$	$G_0^5$	$E_0^5$
$CM_1^1$	$P_1^1$	$CM_1^2$	$\lambda_1^1$	$G_1^1$	$E_1^1$
$CM_1^2$	$P_1^2$	$CM_1^1$	$\lambda_1^2$	$G_1^2$	$E_1^2$
$CM_2^1$	$P_2^1$	0	$\lambda_2^1$	$G_2^1$	$E_2^1$

$$P_0^4 = [0 \quad 1 \quad 1 \quad 0] \quad (38)$$

$$P_1^1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad (39)$$

$$P_2^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (40)$$

Once all elementary agents are created, the  $O$  part in these agents will periodically analyze the relationship information stored in  $K$  (the relationship matrix  $P$ , the  $CAL$ ) to verify whether the

composition/decomposition conditions are met. The composition and decomposition process and conditions are explained in the following section.

### 4.3 Composition / decomposition mechanisms

In different phases of CM lifecycle, a product may work alone or cooperate with other CMs. There are different composition conditions varying over the lifecycle. At the beginning (within a single concrete), the aggregation condition is triggered if a node detects another node in its communicating range based on the communication relationship matrix  $P$ . In the phase of construction or exploitation, the aggregation mechanism can be triggered when the agent of one product detects the arrival of a cooperative agent of another product, or when it receives a cooperation request from users. For example, once the elementary agents of element  $I$  are created as shown in Figure 63, the agent  $CM_0^4$  can invite other elementary agent in its CAL to create an aggregated agent  $CM_1^1$ .

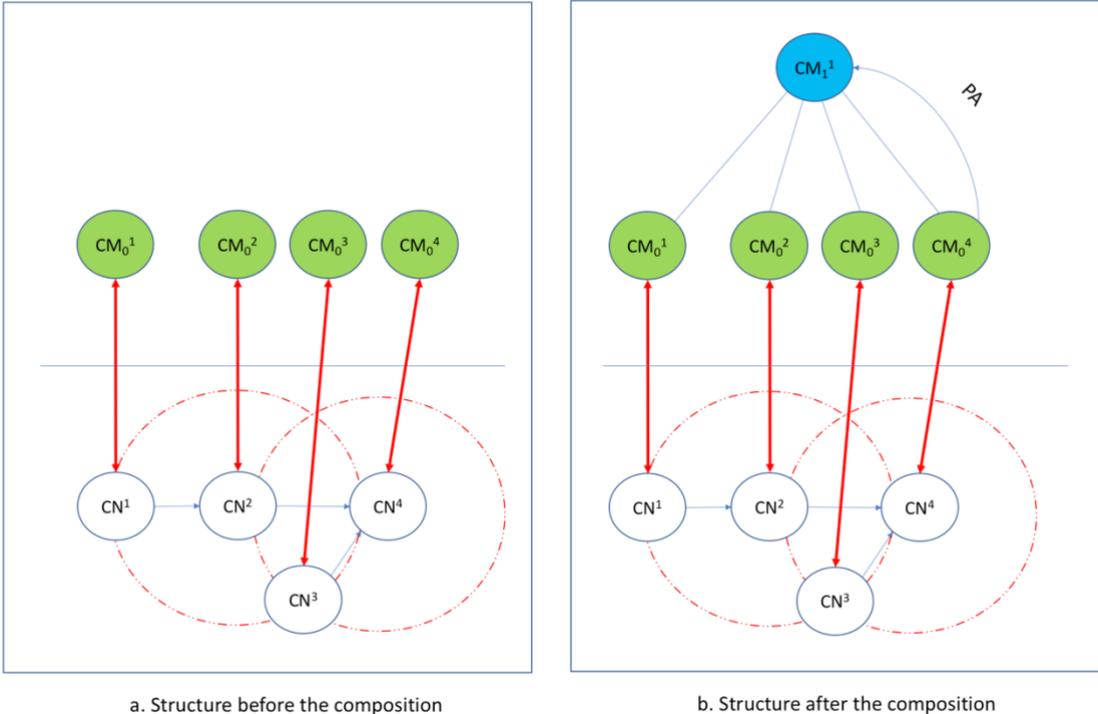


Figure 63 Building recursive architecture with composition process

The sequence diagram for generating parent agent  $CM_1^1$  by the agent  $CM_0^4$  is illustrated in Figure 64. When the observer part of agent  $CM_0^4$  finds that there are some other cooperative agents out in its communicating range (based on its relationship matrix  $P$ ), it applies the  $PA$

operator (composition process) to create the aggregated agent  $CM_1^1$ . The composition process can be explained in five steps:

- 1) The  $CM_0^4$  creates a parent agent  $CM_1^1$  and adds its communication relationship, aggregation coefficients  $(P, \lambda)$  to agent  $CM_1^1$  (message 1 in Figure 64).
- 2) *Request\_compose* messages are sent by  $CM_0^4$  to cooperative agents based on the CAL (message 2 and 3). The invited agents process the composition request and also invites their cooperative agents to join. Such as the invitation sent from the  $CM_0^2$  to  $CM_0^1$  (message 4 in red color) which is not in the CAL of  $CM_0^4$  due to communication range (as shown in Figure 63). All invited agents then reply an *accept\_compose* message or a *refuse\_compose* message directly to the agent  $CM_0^4$  (message 5, 6 and 7). For example, the agents belonging to another group may refuse the composition request. In the case of agreement, their communication relationship and aggregation coefficients information  $(P, \lambda)$  are included in the *accept\_compose* message.
- 3) The agent  $CM_0^4$  adds the relationship matrix and energy information of the accepted agents to its parent node (message 8).
- 4) The agent  $CM_0^4$  then sends *update\_compose* messages to all child nodes. Once received the *update\_compose* message, the agents update their own CAL to ensure they have the same parent agent.

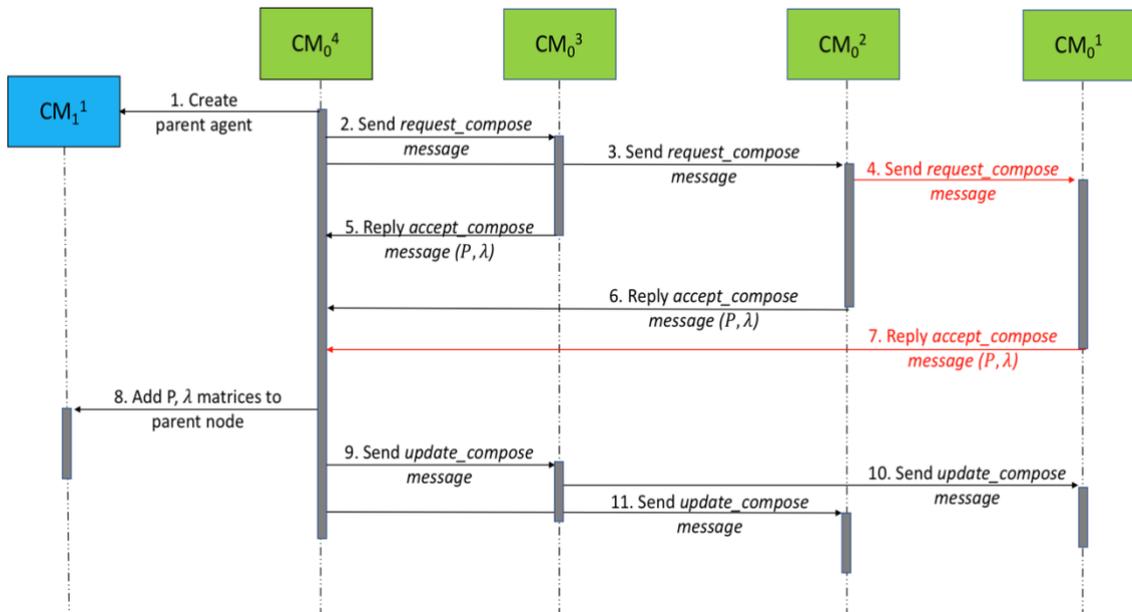


Figure 64 The sequence diagram of composition

Once the aggregated agent  $CM_1^1$  is created and the structure is stable, this aggregated agent has the global view of the whole network with a complete relationship matrix  $P$  and coefficient

matrix  $\lambda$ . This agent can then compute its gathering matrix  $G$  based on the matrix  $P$  and  $\lambda$ , and then construct the analytical energy model  $E$  which can be used to estimate lifetime of this element. Besides, the aggregated agent will also run  $PA$  operator to create an aggregated agent in upper level if its aggregation conditions are met.

The decomposition will be triggered if the agent in the highest level received a request from users or if one child agent detects the disappearance of its communicating node in physical part. For example, the observer of the agent  $CM_0^2$  notices that its communicating node has run out its energy, the decomposition process will be triggered as following:

- 1) The agent  $CM_0^2$  sends a *request\_reduction* to its parent agent  $CM_1^1$ .
- 2) The agent  $CM_1^1$  processes the request and updates its relationship matrix  $P$ , *update\_reduction* messages are then sent to all children.
- 3) The child agents update their own relationship matrix based on the received *update\_reduction* message.

Besides, there are two decomposition possibilities for an aggregated CM which is composed by two aggregated CMs such as  $CM_2^1$  is composed by  $CM_1^1$  and  $CM_1^2$  in the Figure 65. The first decomposition condition (case 1 in the Figure 65) is that the aggregated agent  $CM_2^1$  may receive the decomposition order from the server (the users).

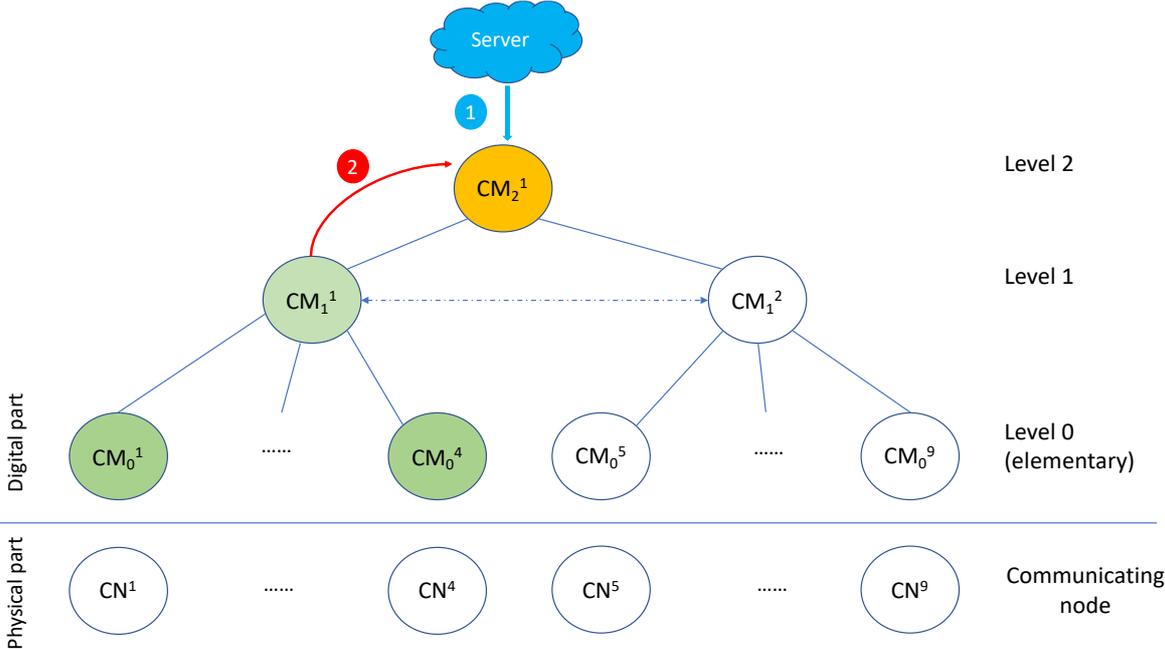


Figure 65 Decomposition for aggregated CM

There are three phases for the decomposition in the first case:

- 1) The agent  $CM_2^1$  broadcasts *request\_stop* to all children agents  $CM_1^1$  and  $CM_1^2$  to stop their activities and remove their parent relationship with  $CM_2^1$ . The *ACK\_stop* messages are sent back to the agent by its children.
- 2) Once receives *ACK\_stop* messages, the agent  $CM_2^1$  kills itself.
- 3) The aggregated agents  $CM_1^1$  and  $CM_1^2$  then evaluate and select suitable data gathering strategies and update their relationship matrices  $P_1^1$  and  $P_1^2$ . These matrices are then transferred to their children to update their strategies with *update\_gathering* messages.
- 4) The *ACK\_gathering* messages are then sent back to the aggregated agents  $CM_1^1$  and  $CM_1^2$  to ensure all agents have updated the new data gathering strategies.

The second decomposition case is that a child agent sends the decomposition request (the agent  $CM_1^1$  in Figure 65). This decomposition process will occur when the agent finds communicating nodes in physical part have run out of energy. Compared with the previous case, there are one more phase: the agent  $CM_1^1$  sends a *request\_reduction* to its parent agent  $CM_2^1$  to begin the decomposition process. Then the decomposition process continues as that in the previous case. At the end, each aggregated agents in level 1 ( $CM_1^1$  and  $CM_1^2$ ) will select their new data gathering strategies.

#### 4.4 Interaction between agents for lifetime estimation

Once composition process and energy model construction are completed, the agent  $CM_2^1$  at the highest level has the global view of all members. Based on the data gathering matrix  $G$  and energy levels, the agent  $CM_2^1$  can estimate residual lifetime of its members. To ensure the accuracy of lifetime estimation, this agent  $CM_2^1$  can ask current energy level of child agents to estimate CM lifetime. The estimation process example is shown in Figure 66.

The agent  $CM_2^1$  first sends *request\_energy* messages to its child agents ( $CM_1^1$  and  $CM_1^2$ ). These agents then check whether they have current energy level information. They can reply directly to their parent node if they have (such as the aggregation agent  $CM_1^2$  in this example). Otherwise, the agent ( $CM_1^1$ ) retransfers the request message to all its child agents to get their energy information (this interaction transformation is the *PI* operator). Once received all replies, the composed agent ( $CM_1^1$ ) can then aggregate energy information of its members and reply an

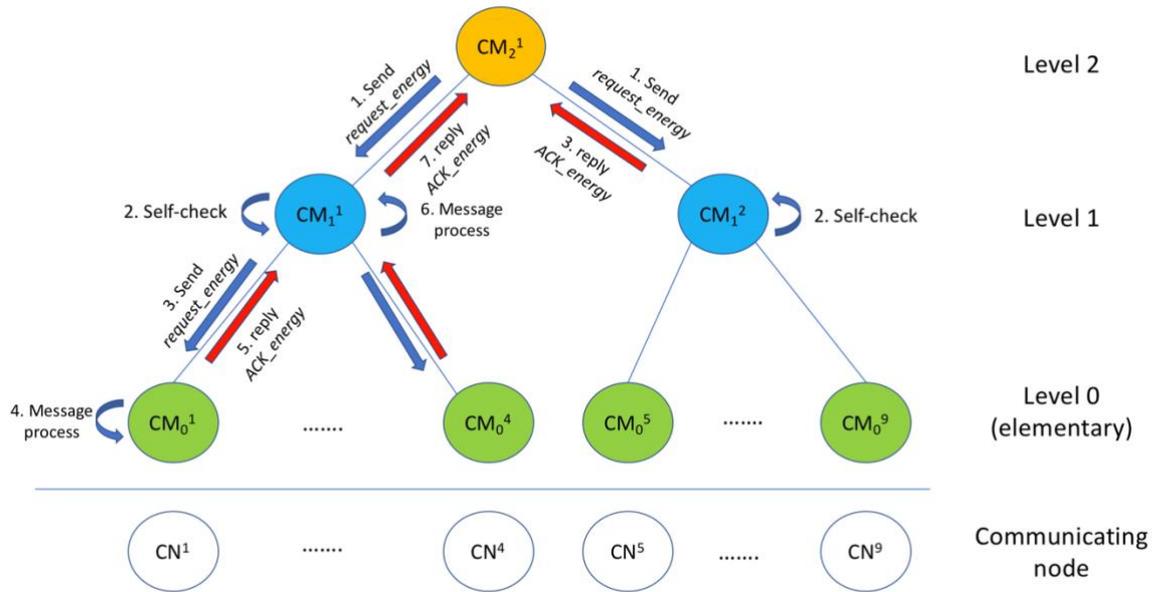


Figure 66 CM lifetime estimation process

aggregated *ACK\_energy* message (minimal or average energy level) to parent agent  $CM_2^1$  thanks to the *PI* operator. At the end, the agent  $CM_2^1$  can then estimate CM lifetime based on the energy consumption model and the recent energy levels of its members.

#### 4.5 Cooperation between two aggregated elements

In some phases such as building construction, two aggregated CM products (by example, two individual concretes in Figure 67) may cooperate with each other to improve energy performance. These elements can be implemented in a multi-agent simulator NetLogo where each communicating node is represented by an agent (the simulator NetLogo will be presented in Chapter 5).

During the aggregated agent’s lifecycle, its child agents check periodically whether they have new relationship possibilities or if new element arrives. In this example in Figure 67, the agent  $CM_0^9$  detects the agent  $CM_0^4$ . It then runs the cooperation process as shown in Figure 68 to decide the cooperation way. At the beginning, the agent  $CM_0^9$  sends a *warning* message to parent agent  $CM_1^2$  (message 2). A *request\_coop* message that includes energy information ( $E$ ) and relationship matrix( $P$ ) of this element is sent back to  $CM_0^9$  (message 3). The request is then transferred to the aggregated agent ( $CM_1^1$ ) via the connected agent ( $CM_0^4$ ) (message 4 and 5). Once the evaluation of cooperation is computed, an *ACK\_coop* message is sent back. There are

three cooperation possibilities (*Case 1*: refuse; *Case 2*: association and *Case 3*: aggregation) in the *ACK\_coop* message. These cooperation possibilities are discussed hereafter.

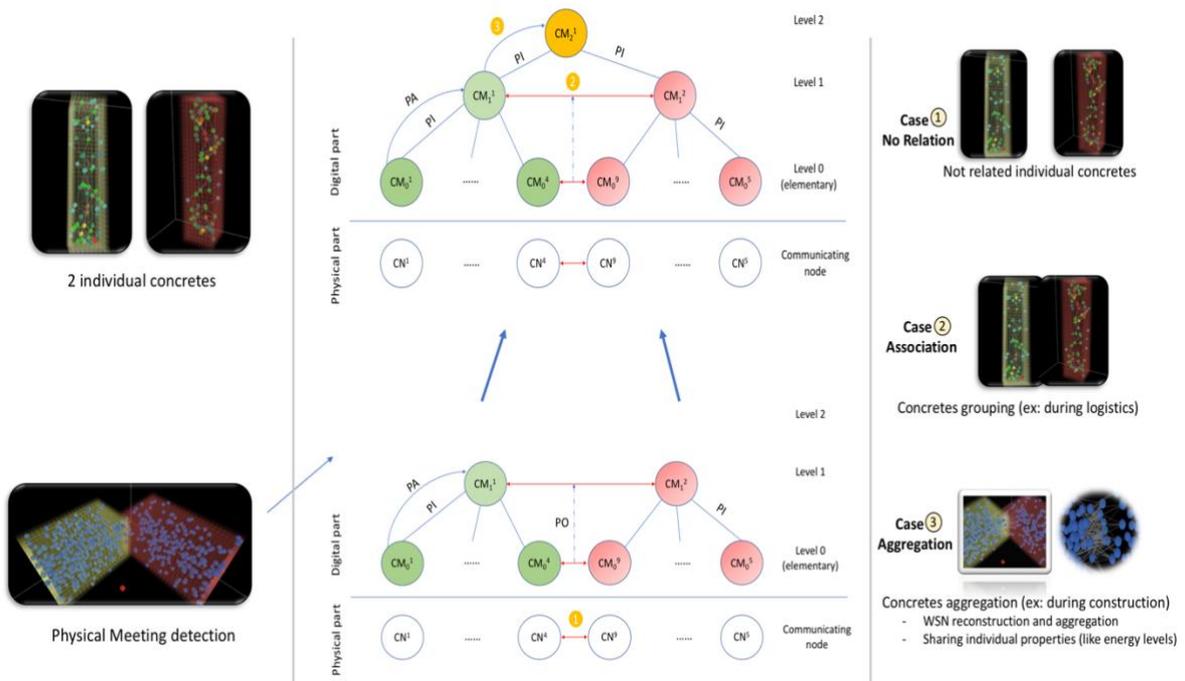


Figure 67 Cooperation of two elements

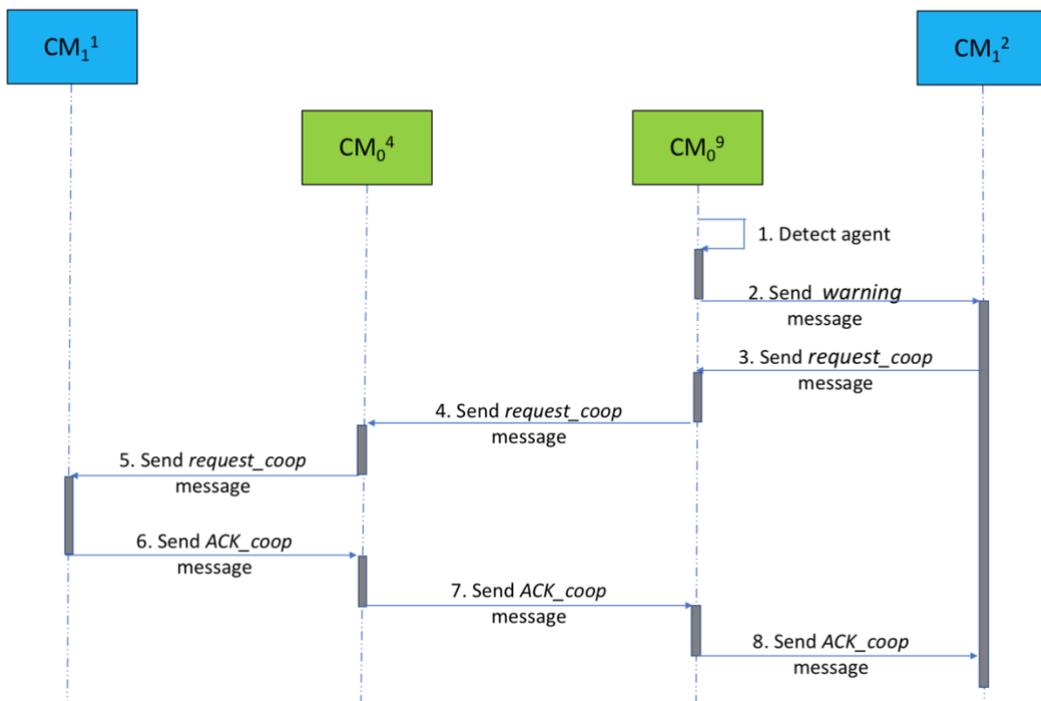


Figure 68 Cooperation process for a new element

The first case is no relation. Once the elementary agents receive a *refuse\_coop* message, they maintain the original working state as shown in Figure 69. Therefore, there are no changes to the communication relationship matrices and the aggregation matrices.

The second case is association. If the parent agent confirms that the new element is related, it then replies to all children with the association messages. The aggregated agents can then build the relationship with each other as shown in Figure 70. This relationship transformation from the lower level to the high level is the *PO* operator. In this case, these two parts can always know each other's status and cooperate to accomplish their common tasks.

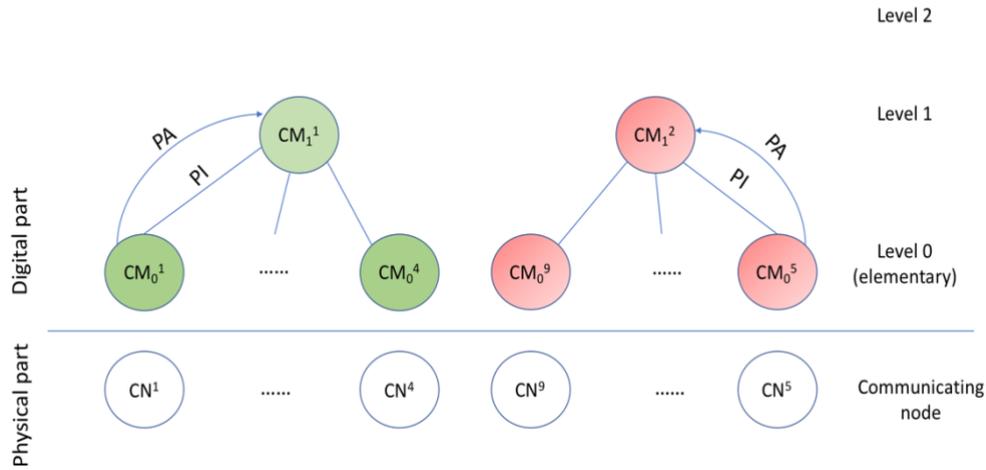


Figure 69 The first cooperation case: independent

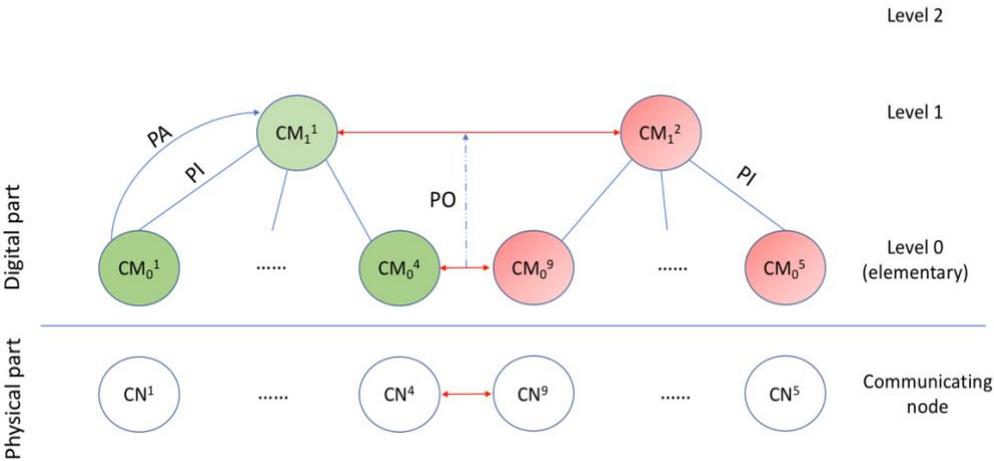


Figure 70 The second cooperation case: association

The third case is aggregation. When the recursive conditions are met, all elementary agents stop their activity and wait orders from their parent node. The composed agent ( $CM_I^2$ ) then begins the composition process thanks to the  $PA$  operator to build the agent in level 2 as shown in Figure 71.

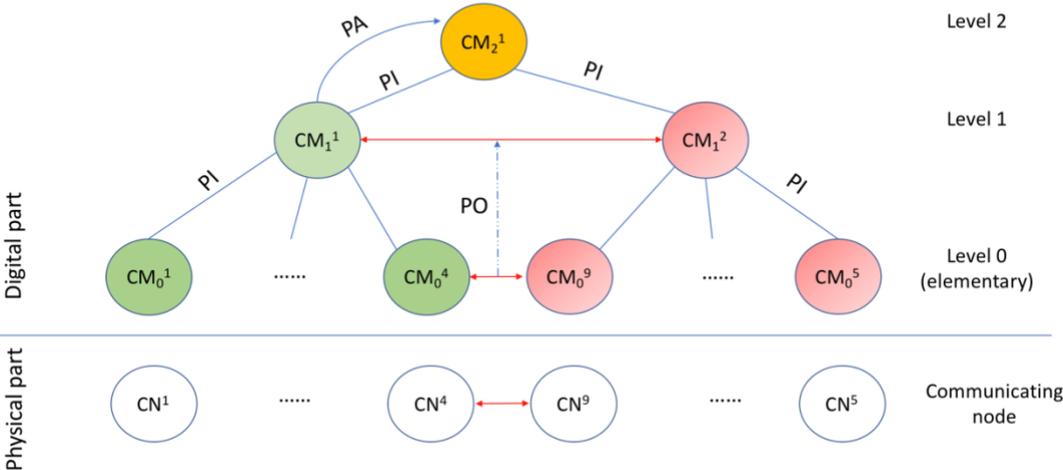


Figure 71 The third cooperation case: aggregation

In this case, the agent in the highest level can evaluate available data gathering strategies based on the network information (relationship matrix  $P$ , gathering matrix  $G$ ) and energy information (residual energy  $E$ ) of its members to select a suitable routing protocol (new gathering matrix  $G$ ). This new strategy will then be transferred to all elementary agents. For example, the element 2 shares the cluster leader to the element 1, and only the cluster leader in element 2 will send information to the  $BS$ .

### 4.6 Energy consumption estimation scenarios

With the formalized recursive architecture, composition and decomposition mechanisms, some application scenarios based on the proposed analytical energy model in *chapter 3* are discussed in this section. We would like to analyze energy consumption of nodes in the previous example (two independent elements). Node 4 and node 9 are designated as root node and cluster leader of their respective element. The recursive architecture for these two elements is illustrated in Figure 72 where the aggregated agents  $CM_I^1$  and  $CM_I^2$  have the global view of their members respectively. Energy consumption is discussed for data collection in independent communicating concretes and in the case of the two elements aggregation hereafter.

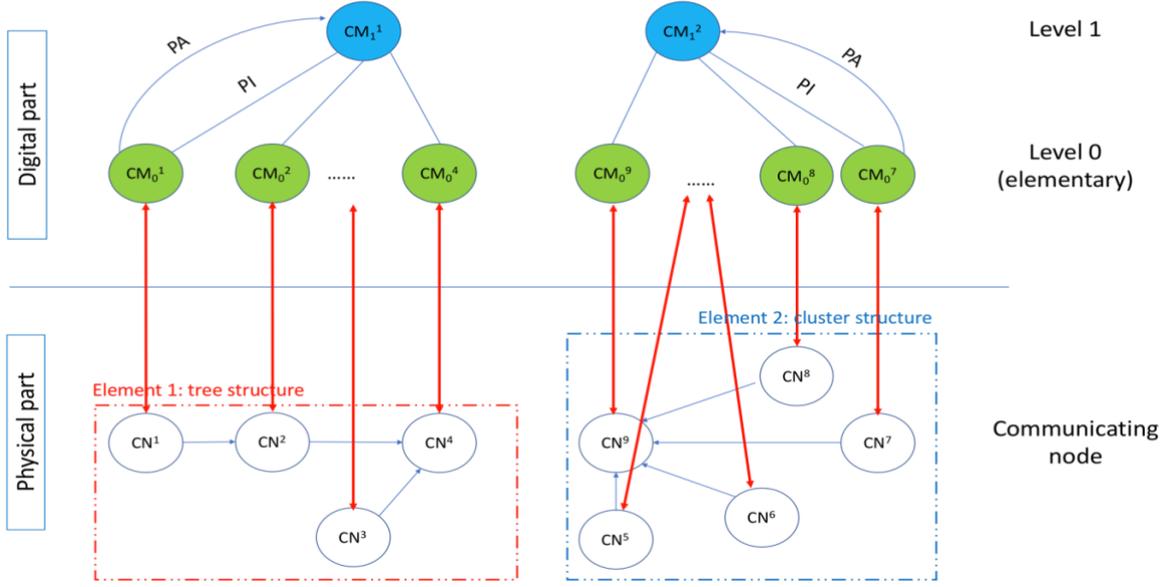


Figure 72 Data collection in independent concrete

The parameter assumptions for energy consumption estimation are given in Table VI. The size of minimal data packet  $S_{min}$  and maximal data packet  $S_{max}$  are 40 bytes and 100 bytes, their transmission duration are  $\alpha_{min}$  (20 ms) and  $\alpha_{max}$  (50 ms) respectively. The IFS  $\beta$  is 10 ms. The size of ACK packet and of a unique monitoring data packet is 40 bytes, therefore, the time for transmitting a acknowledgment message is also 20 ms. The synchronization duration  $d_{syn}$  is set to 10 ms for each node.

Table VI. Parameter assumptions

Parameters	Value	Unit
$\alpha_{min}$	20	ms
$\alpha_{max}$	50	ms
$\beta$	10	ms
$\gamma$	20	ms
$S_{min}$	40	byte
$S_{max}$	100	byte
$d_{syn}$	10	ms

#### 4.6.1 Data collection with lossy data aggregation

During the composition process, the relationship matrix, aggregation coefficients will be added to the created aggregated agent as mentioned above. Figure 73 illustrates the construction of relationship matrix at the aggregated agent  $CM1^1$ . In this example, the agent  $CM0^4$  creates the

aggregated agent  $CM_I^1$  and invites cooperative agents to join, all accepted agents reply the  $accept\_compose$  messages which includes the relationship matrix  $P$ . At the end, the agent  $CM_0^4$  add all received relationship matrices and its own to the aggregated agent  $CM_I^1$  which then has the global view of all members.

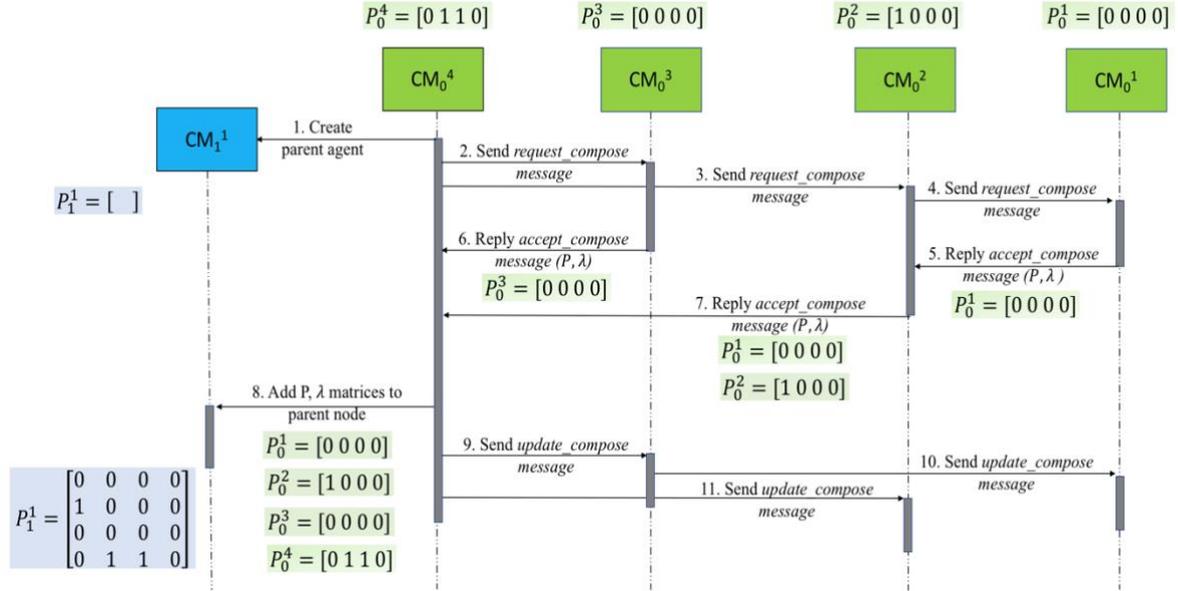


Figure 73 Composition process for the construction of relationship matrix

The aggregation coefficients  $\lambda$  are retrieved via the same process. The aggregated  $CM_I^1$  can then compute the gathering matrix  $G$  and obtain the energy consumption model  $E$  for each node. The aggregation coefficients depend on the applicative needs. For example, with lossy data aggregation, the aggregation coefficients of a node  $i$  with  $n$  children are  $1/(n+1)$ . After the composition process is completed, the aggregation coefficient matrices  $\lambda_1^1$  and  $\lambda_1^2$  of the two elements can be presented as in (41) and (42). With aggregation coefficients and with the relationship matrices, the gathering matrices  $G_I^1$  and  $G_I^2$  can then be computed as in (43) and (44).

$$\lambda_1^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \end{bmatrix} \quad (41)$$

$$\lambda_1^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix} \quad (42)$$

$$G_1^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1/6 & 1/6 & 1/3 & 1/3 \end{bmatrix} \quad (43)$$

$$G_1^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix} \quad (44)$$

With the gathering matrices, the size of transferred and received messages for these two elements can be computed at the aggregated agents as follows:

$$M_{tx_1}^1 = G_1^1 \times S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1/6 & 1/6 & 1/3 & 1/3 \end{bmatrix} \begin{bmatrix} 40 \\ 40 \\ 40 \\ 40 \end{bmatrix} = \begin{bmatrix} 40 \\ 40 \\ 40 \\ 40 \end{bmatrix}$$

$$M_{tx_1}^2 = G_1^2 \times S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix} \begin{bmatrix} 40 \\ 40 \\ 40 \\ 40 \\ 40 \end{bmatrix} = \begin{bmatrix} 40 \\ 40 \\ 40 \\ 40 \\ 40 \end{bmatrix}$$

$$M_{rx_1}^1 = P_1^1 \times M_{tx_1}^1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 40 \\ 40 \\ 40 \\ 40 \end{bmatrix} = \begin{bmatrix} 0 \\ 40 \\ 0 \\ 80 \end{bmatrix}$$

$$M_{rx_1}^2 = P_1^2 \times M_{tx_1}^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 40 \\ 40 \\ 40 \\ 40 \\ 40 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 160 \end{bmatrix}$$

With these scenario parameters, the active duration of nodes can be calculated at all nodes. As the root node or cluster head always consumes more energy than other structure members, the active duration for the root node (node 4) is then analyzed in (45). A more complete energy consumption calculation for all nodes involved in these two elements is given in *Chapter 5*.

$$D_{active}(4) = D_{rx}(4) + \beta + D_{tx}(4) \quad (45)$$

Where,

$$D_{tx}(4) = \left\lfloor \frac{M_{tx}(4)}{S_{max}} \right\rfloor * \alpha_{max} + \max \left( \alpha_{min}, \left\lfloor \frac{M_{tx}(4) \bmod S_{max}}{S_{min}} \right\rfloor * \alpha_{min} \right)$$

$$\begin{aligned}
&= \left\lfloor \frac{40}{100} \right\rfloor * 50 + \max \left( 20, \left\lfloor \frac{40 \bmod 100}{40} \right\rfloor * 20 \right) \\
&= 0 + 20 = 20ms
\end{aligned}$$

$$\begin{aligned}
D_{rx}(4) &= d_{syn} + \sum_{\forall j, P(i,j)=1} d_{tx}(j) + \beta + \gamma \\
&= d_{syn} + d_{tx}(2) + \beta + \gamma + d_{tx}(3) + \beta + \gamma \\
&= 10 + 20 + 10 + 20 + 20 + 10 + 20 \\
&= 110
\end{aligned}$$

$$E_1^4 = T * P_{sleep} + D_{active}(4) * (P_{active} - P_{sleep})$$

The transmitted message size at the *node 4* is *40 bytes*, therefore, it sends data in a small transmission frame (transmission duration is *20 ms*). For each iteration, its active duration is the sum of the time for message transmission  $D_{tx}(4)$ , the time for message reception  $D_{rx}(4)$  and an IFS ( $\beta$ ). With the active duration, energy consumption of this node can then be computed. In this section, we just illustrate how to apply the analytical estimation model to compute active duration and energy consumption. In the *section 5.2*, energy consumption of both the chain leader and its members will be analyzed and compared.

#### 4.6.2 Energy consumption estimation for the composition of elements

There are three cooperation modes as mentioned in the previous section: independent, associate and aggregation. Compared with the first two modes, the last one, the aggregation of two elements, is the most complex case, where both cooperation and aggregation mechanisms are triggered. Such as the aggregation of two elements in Figure 74, the nodes in the element *1* transmit data to the BS via the chain leader of element *2* (the orange link between the node *4* and the node *9*). In this example, we suppose that lossy data aggregation is used in these two elements.

There are three steps for this cooperation. First, the agent in the first element detects the arrival of the cooperative element and runs the cooperation process to decide the cooperation way (see Figure 68). Since the cooperation way is aggregation, the aggregated agent of element *2* can then apply the *PA* operator to build an aggregated agent in the upper level (see Figure 71). The created agent  $CM_2^1$  will have the global view of all members, the relationship matrix  $P_2^1$  as well as the aggregation coefficient matrix  $\lambda_1^1$  will be constructed as in Figure 73. Third, once the

composition process is done, the aggregated agent  $CM_2^1$  in the level 2 then broadcast the new data collection strategies to all agents via the  $PI$  operators.

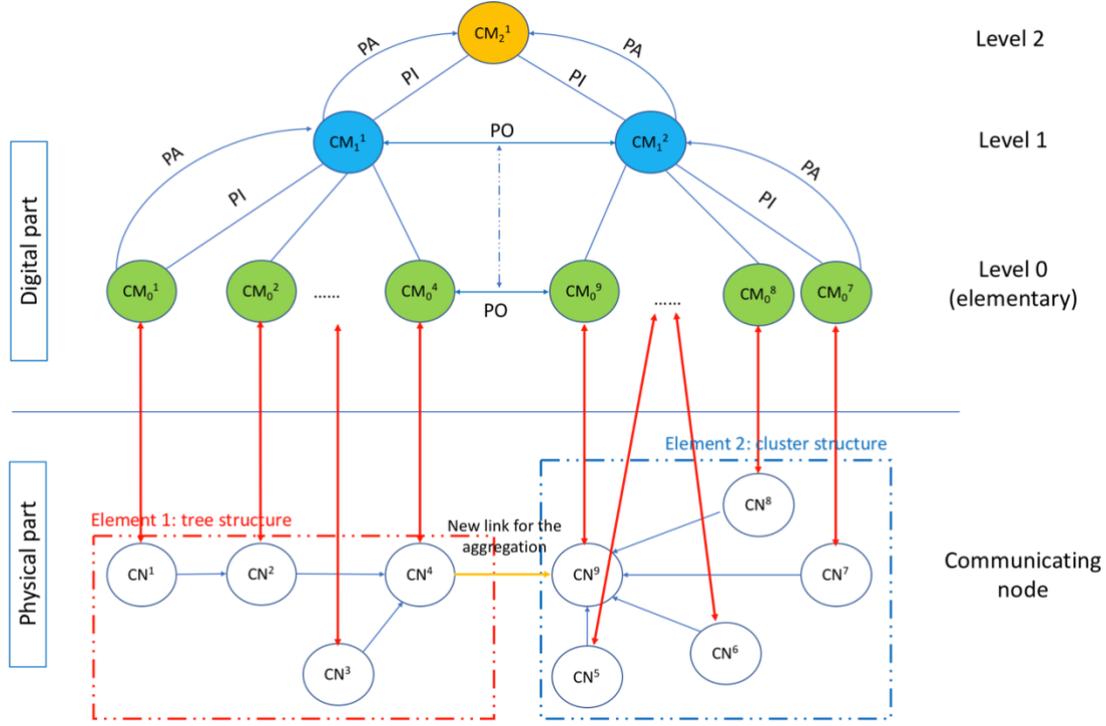


Figure 74 Aggregation of two elements

The new relationship matrix  $P_2^1$  and coefficient matrix for the aggregated agent  $CM_2^1$  is shown as in (46) and (47). Based on those two matrices, the gathering matrix  $G_2^1$  can be computed at the agent  $CM_2^1$  as in (48) which can be seen as the fusion matrices as the (37) in section 3.3.3.

$$P_2^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (46)$$

$$\lambda_2^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ & & & & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{bmatrix} \quad (47)$$

$$G_2^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1/36 & 1/36 & 1/18 & 1/18 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{bmatrix} \quad (48)$$

With the relationship matrix and the gathering matrix, the transmitted message size and the received message size can be computed as follows:

$$M_{tx_2}^1 = G_2^1 \times S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1/36 & 1/36 & 1/18 & 1/18 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{bmatrix} \begin{bmatrix} 40 \\ 40 \\ 40 \\ 40 \\ 40 \\ 40 \\ 40 \\ 40 \\ 40 \end{bmatrix} = \begin{bmatrix} 40 \\ 40 \\ 40 \\ 40 \\ 40 \\ 40 \\ 40 \\ 40 \\ 40 \end{bmatrix}$$

$$M_{rx_2}^1 = P_2^1 \times S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 40 \\ 40 \\ 40 \\ 40 \\ 40 \\ 40 \\ 40 \\ 40 \\ 40 \end{bmatrix} = \begin{bmatrix} 0 \\ 40 \\ 40 \\ 80 \\ 0 \\ 0 \\ 0 \\ 0 \\ 200 \end{bmatrix}$$

The transmitted message size is the same for all nodes with lossy method. However, the more the children, the more the received messages. Therefore, the active duration of the leader (the node 9) can be computed as follows:

$$D_{active}(9) = D_{rx}(9) + \beta + D_{tx}(9) + d_{syn}(9)$$

Where,

$$\begin{aligned} D_{tx}(9) &= \left\lfloor \frac{M_{tx}(9)}{S_{max}} \right\rfloor * \alpha_{max} + \max \left( \alpha_{min}, \left\lfloor \frac{M_{tx}(9) \bmod S_{max}}{S_{min}} \right\rfloor * \alpha_{min} \right) \\ &= \left\lfloor \frac{40}{100} \right\rfloor * 50 + \max \left( 20, \left\lfloor \frac{40 \bmod 100}{40} \right\rfloor * 20 \right) \\ &= 20ms \end{aligned}$$

$$\begin{aligned} D_{rx}(9) &= d_{syn} + \sum_{\forall j, P(i,j)=1} d_{tx}(j) + \beta + \gamma \\ &= d_{syn} + d_{tx}(4) + \beta + \gamma + d_{tx}(5) + \beta + \gamma + d_{tx}(6) + \beta + \gamma + d_{tx}(7) + \beta + \gamma + d_{tx}(8) + \beta + \gamma \end{aligned}$$

$$\begin{aligned}
&= 10 + 5 * (20 + 10 + 20) \\
&= 260
\end{aligned}$$

$$E_1^9 = T * P_{sleep} + D_{active}(9) * (P_{active} - P_{sleep})$$

Compared with the independent case, leader (node 9) consumes more energy for data collection in this example of cooperation. However, the energy consumption of the node 4 decreases, because it does not need to directly transmit data to the BS. In this case, the CM agent in the highest level has the global view of the communication relationship and energy information of all members. This agent can evaluate and select suitable data collection solution to limit and balance the energy spent by the real Communicating Nodes and so extend CM lifetime.

## 4.7 Conclusion

In this section, a new recursive agent architecture for the CM, adapted from the MAS-R agent model, is introduced. The communication relationship matrix, aggregation coefficients and all parameters to build the analytical energy consumption model are added to the agent model. Three operators *PA*, *PI* and *PO* are used to build the dynamic recursive architecture. The composition and decomposition conditions are detailed in section 4.3. The example of interaction between agents for *PI* operator is presented in section 4.4. Besides, different cooperation solutions between elements are presented in section 4.5 as well as the *PO* operator for transforming relationship of agents from low level to upper level. The application for energy consumption estimation with proposed analytical models is also discussed in the case of data collection within single element and cooperation between elements in section 4.6.

In the proposed recursive architecture, the agent in the highest level has the global view of all members (the communication relationship, gathering matrix, energy level and so on). This agent can adjust data collection strategies to meet environment changes (such as the change of product lifecycle phase). In the following chapter, the accuracy of the analytical energy consumption model is analyzed with data collection test in a physical platform (*section 5.1*). The proposed recursive architecture and composition mechanisms will be implemented and verified via a multi-agent simulator (see section 5.2).



# Chapter 5

## Experiments, developments and validation

### Contents

<a href="#">5.1 Validation of the energy consumption model for a chain structure</a> .....	109
<a href="#">5.1.1 Chain-based data collection platform</a> .....	110
<a href="#">5.1.2 Energy consumption of Arduino nodes with XBee shield</a> .....	111
<a href="#">5.1.3 Data collection in 3-node chain</a> .....	113
<a href="#">5.1.4 Data collection in 6-node chain</a> .....	115
<a href="#">5.2 Validation of recursive architecture</a> .....	117
<a href="#">5.2.1 Analysis of energy consumption within a single communicating concrete</a> .....	119
<a href="#">5.2.2 Using the simulator to analyze the First Node Death time</a> .....	122
<a href="#">5.2.3 Cooperation of communicating concretes</a> .....	125
<a href="#">5.3 Analysis</a> .....	130

To address the research issues for the realization of the communicating concrete, energy saving solutions and analytical energy estimation models have been proposed in chapter 3. Dedicated agents have also been defined as well as the recursive architecture in chapter 4. As consequence, there are two main experimental steps before implementing these proposals into real concrete pieces: first, the energy evaluation of a chain structure to show that our predictive models are reliable and have a high accuracy (*section 5.1*); second, implementation of the proposed recursive architecture in a multi-agent simulator to verify the usefulness of proposed composition and cooperation process, and illustrate its benefits for energy estimation in the cases of cooperation (*section 5.2*).

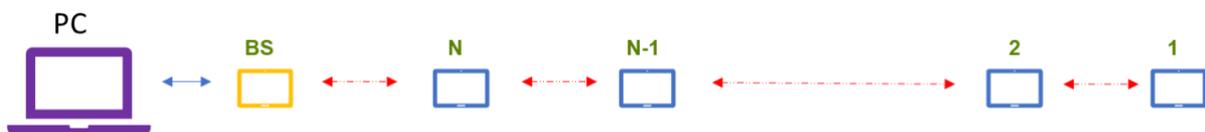
### 5.1 Validation of the energy consumption model for a chain structure

In the first part of this section, the materials used for our testbed are introduced. Then, the first experimental step deals with the measurement of power consumption in active and sleep modes. In a second step, energy consumptions with/without aggregation are compared to validate the aggregation option with a 3-node chain structure. The real energy consumptions of 6 nodes in chain are recorded and compared to verify the accuracy of our energy estimation model.

### 5.1.1 Chain-based data collection platform

The chain-based routing structure is suitable for a small network as that within one McBIM element. Before applying it to a real concrete element, a physical platform is designed to verify the accuracy of the proposed energy estimation model.

The design of the chain structure is illustrated in Figure 75. This platform is constituted of microprocessor chips Arduino Uno equipped with XBee communicating module as shown in Figure 76. c. Each node owns an  $ID$  between  $1$  and  $N$ , from the furthest node to the BS. The last node is connected to a PC via a USB cable that can not only transfer data but also provide the necessary power. The other nodes are supplied via standard  $9v$  batteries. Data collection begins from the furthest nodes until the BS. The PC shows all received messages.

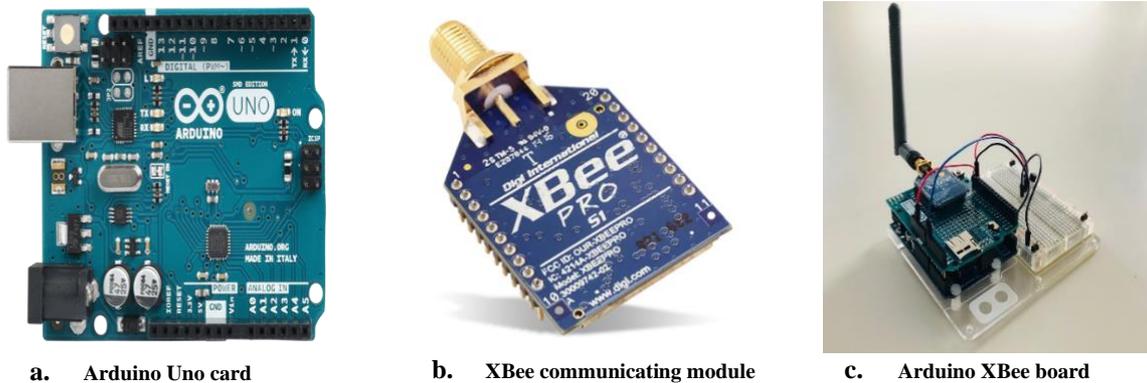


*Figure 75 Chain-based data collection*

The microprocessor chips Arduino Uno is based on a Microchip AT mega 328P microcontroller. This chip can be powered by USB cable or an external battery between  $7$  to  $20\text{ v}$  ( $9\text{ v}$  is recommended). It has  $14$  digital pins and  $6$  analog pins. In addition, the Arduino IDE (Integrated Development Environment) makes programming and compilation simpler for users with C language. The *ATmega328* provides a pair of communication port (the *Rx* and *Tx*) that enables the communication between the card with computer or other boards. Moreover, a lot of sensors can be used such as temperature sensors, humidity sensors, light sensors and so on. Its various external devices and basic computational capability allow its use in different areas. In our experiment, this board is used with a XBee communicating module to achieve wireless data collection.

The XBee *S1* is a wireless communication module that uses a  $2.4\text{ GHz}$  transceiver (IEEE 802.15.4) to communicate with other XBee modules. It has two versions: XBee and XBee Pro. These modules have  $20$  pins and work between  $2.8$  to  $3.4\text{ Voltage}$ . Compared with a short indoor communication distance ( $30$  meters) of XBee, the XBee pro modules have greater capabilities. Its maximum transmission range can reach  $90$  meters. Meanwhile, it consumes

more energy. An Arduino XBee shield is used to enable the communication between XBee and Arduino.



*Figure 76 The materials of data collection platform*

To ensure a high-quality service, the XBee Pro modules are used for our experiment. These modules work in five modes: Idle mode (waiting to enter other modes), transmit mode (sending messages), receive mode (reading messages) and sleep mode (waiting to wake up with a low-power consumption). Using sleep mode could save energy and extend the node lifetime. However, to efficiently use sleep mode, synchronization of nodes is required.

### 5.1.2 Energy consumption of Arduino nodes with XBee shield

The power consumption of the Arduino XBee board card is measured periodically. This board card can be powered either by USB cable or external battery. Measuring power consumption with battery may be affected by its surrounding environments and the result may be instable.



*Figure 77 USB wattmeter*

Therefore, a USB wattmeter designed in our laboratory is used, as shown in Figure 77. This wattmeter has three cables, one connected to the board, the two others are connected to a computer. A software tool is also provided to show the measured voltage, current and power of the electronic board. The maximum supported voltage and current are respectively 5 volts and 2 amperes. The active state contains all modes (idle; transmit; receive) except the sleep mode. During sleep mode, the XBee module does not receive any information from the others and stay in low-energy state. It can be reactivated for new transmission. The sleep mode of XBee module is controlled by the Arduino Uno.

In order to verify the usefulness of sleep mode for data collection, a simple program is implemented on Arduino XBee card 1 for two tests. During the first test, the *node 1* sends a simple letter ‘A’ every five seconds for twelve cycles, but always remains in active state (it only enters in idle mode after transmitting its message). In the second one, the node lets its XBee module enter into sleep mode after transmission. The energy power for these two tests is shown in Figure 78.

It is clear that the node in the first case consumes more energy. In the second one, its power is reduced when the XBee module enters into sleep mode. The total energy consumption is around  $36.89 J$  for the first test. The duty cycle of the XBee module (defined as the ratio active time/period) is  $100\%$ . However, the energy cost is reduced to  $27.74 J$  using sleep mode which is only  $75.2\%$  of that in the first test. The duty cycle of radio module is  $18.3\%$  with a total active time of  $11.90 s$  (almost one-fifth of the former) and the gains of energy consumption reaches  $63\%$ . This proves that using sleep mode in WSN can significantly extend the network lifetime.

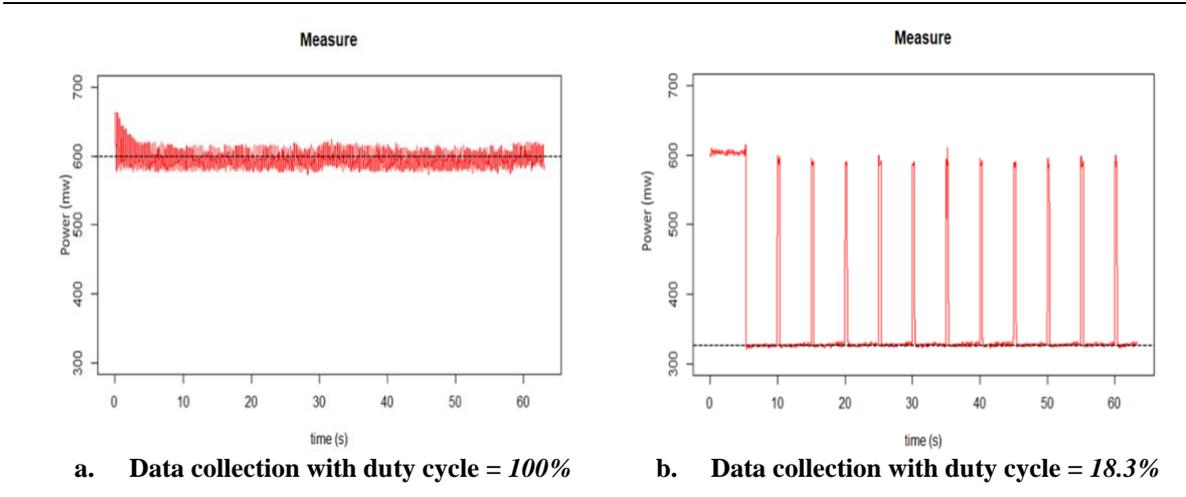


Figure 78 Radio module sleep mode test

The average power of a node in active state is around  $600\text{ mw}$ , and  $320\text{ mw}$  in sleep state where the power of Arduino alone (without the XBee shield) is  $275\text{ mw}$ . Therefore, the power of XBee module with its shield in active / sleep mode are  $325\text{ mw}$  and  $45\text{ mw}$  respectively. Besides, the computing ability of nodes could also be used to reduce the data transfer. In the following, a study on chain-based data collection with or without in-network data aggregation is presented.

### 5.1.3 Data collection in 3-node chain

Before applying our approach for a long chain, a short chain structure is first considered to verify whether our synchronization approach works, and the usefulness of in-network data processing at intermediate nodes. A three-node chain is shown in Figure 79.

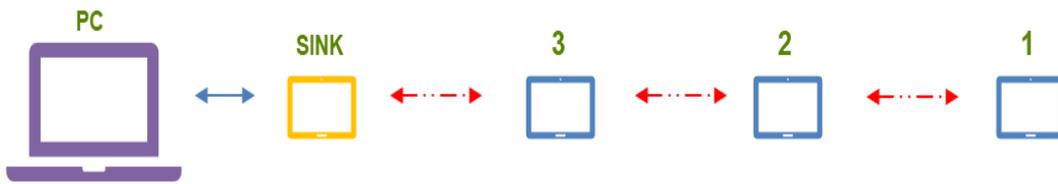


Figure 79 Three-node chain structure

In this structure, data collection process begins from node 1 to node 3. A card is used as sink node (or BS) to collect and upload information to PC. Four hypotheses for the data collection process are considered:

- All nodes (except the farthest node from the BS) employ the same program.
- Each node sends its data (7 bytes for the monitoring of temperature) with the same period  $T$  (5 s).
- Each node lets the XBee module enter into sleep mode after it receives an *ACK* message.
- The BS node is powered by the PC, other nodes are powered with external batteries.

After few synchronization tests with Arduino XBee card, the minimal duration for data transmission ( $\beta$ ) has been estimated around  $20\text{ ms}$ . A long IFS is used to avoid the message collision ( $\alpha$  is set to  $100\text{ms}$ ). To ensure the reception of message, node should wake early before the arrival of message from child node, the  $d_{syn}$  is set to  $10\text{ ms}$ . With these parameters, the energy consumption of the whole network can be estimated thanks to our energy consumption models (Figure 80). The  $x$ -axis and  $y$ -axis represent the chain size and the network energy consumption for each period respectively. This result is a theoretical projection of the energy cost of a chain as a function of the number of nodes. For data collection in a chain structure, the predicted result is parabolic for the case without aggregation and linear with aggregation.

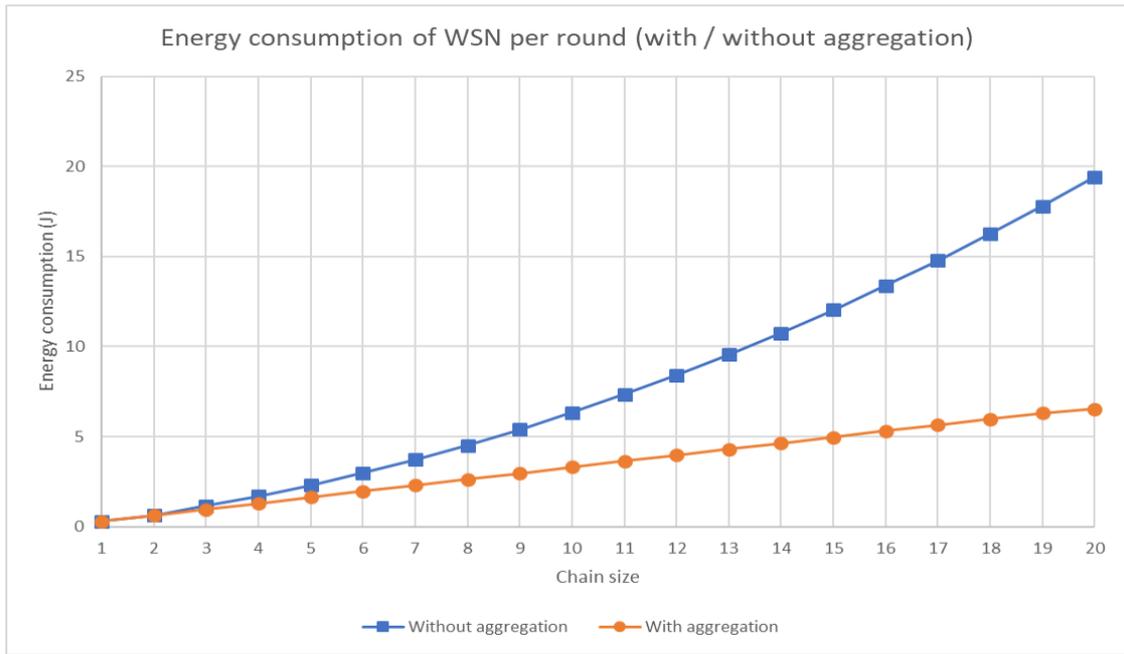


Figure 80 Estimated consumption of the whole chain

Both data collection experiments with / without aggregation are then performed three times during 65 seconds in the 3-node chain. The active times of nodes during data collection **without** aggregation are shown in Figure 81. The active duration of node 1, node 2 and node 3 are in blue, orange and grey color.

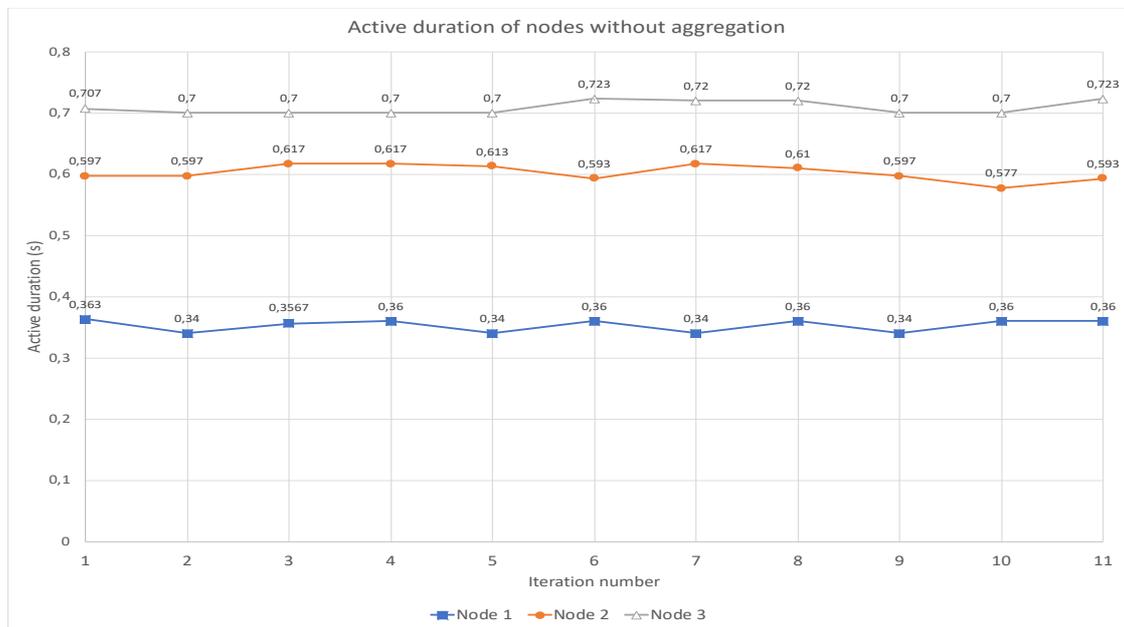


Figure 81 Active duration of nodes during data collection without aggregation

The average measured active duration for node 1, node 2 and node 3 are 0.353 ms, 0.603 ms, 0.708 ms where their estimated values are 0.36 ms, 0.63ms and 0.75ms. From node 1 to node 3, it is obvious that the duration of activity increases. For this test, the hot-spot problem is clear.

The farther the node from the BS, the shorter its active duration. The active duration of *node 3* (nearest node to the BS) is around 2.5 times higher than that of *node 1*. The comparison of the estimation given by our energy model with the real test values leads to prediction errors for *node 1* to *node 3* of 1.98%, 4.48% and 5.93% respectively (errors computed with (49)). The average prediction error is 4.13%.

$$prediction\ error = \frac{abs(Measured\ energy\ cost - Estimated\ cost)}{Measured\ energy\ cost} \quad (49)$$

The active durations for data collection **with aggregation** from node 1 to node 3 are 245 ms, 391 ms and 380 ms respectively, where the estimated values are 240 ms, 390 ms and 390 ms. In this test, the active durations for intermediate nodes (nodes 2 and 3) are almost the same. As a conclusion, the hot-spot problem is avoided by using data aggregation. We also observe that the furthest node works less time than the others. Because it does not have a child, no ACK reply is required. For this experiment, the errors between the real test and our energy estimation model are 2.04%, 0.26%, 2.63% from node 1 to node 3 respectively. The average error is 1.64%. Our estimation model is adapted for both tests.

Duty cycle plays an important role for energy consumption at nodes. The average duty cycles from node 1 to 3 are respectively 7.08%, 12.04%, 14.16% in case of data collection without aggregation. These values are reduced to 4.90%, 7.82% and 7.60 % with data aggregation. Using data aggregation avoids the hot-spot problem for chain-based structure and significantly reduces the energy consumption at node. In next section, this experiment is executed on a 6-node chain structure, to validate our models on longer structures.

#### 5.1.4 Data collection in 6-node chain

The previous experiments show that using sleep mode and aggregation techniques can significantly reduce energy consumption on a three-node chain-based structure. The following experiment aims at comparing the real energy consumption with our proposed energy estimation model for a longer chain structure. Therefore, a six-node chain structure is built as shown in Figure 82.



Figure 82 Six-node chain testbed

We test data collection without aggregation. Hypotheses are the same as in the 3-node experiment. Each node sends its message every five seconds. It enters in sleep mode when it receives an *ACK* message. The comparison of average active duration for the nodes (*node 1 to 6*) is illustrated in Figure 83 where  $\alpha$  (transmission duration),  $\beta$  (*IFS*) and  $d_{syn}$  (Duration for synchronization) are 20 ms, 100 ms and 10 ms. The estimation errors for this 6-node chain are presented in Table VII. The average estimation errors from node 1 to node 6 are respectively: 4.85%, 3.33%, 6.18%, 8.53%, 6.15%, 0.94%. Therefore, the average prediction error for the whole WSN is around 5%.

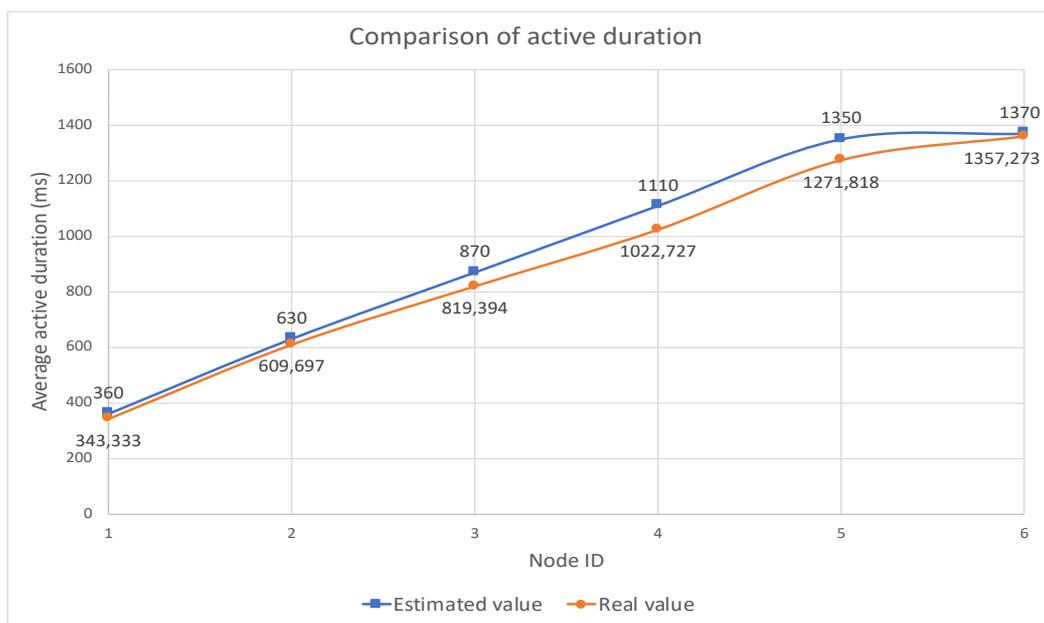


Figure 83 Test of 6-node chain structure

Table VII. Estimation errors of activity durations in six-node chain structure

N_ID	Estimated value (ms)	Measured value (ms)	Absolute error (ms)	Percentage prediction error
1	360	343.333	16.667	4.85%
2	630	609.697	20.303	3.33%
3	870	819.394	50.606	6.18%
4	1110	1022.727	87.273	8.53%
5	1350	1271.818	78.182	6.15%
6	1370	1357.273	12.727	0.94%

Besides, the average error for data collection with aggregation is also similar. Both tests prove the correctness of our energy estimation model for a chain-based structure. This model can give fast and relatively accurate energy cost results and can help to know the embedded WSN lifetime. In the following section, this analytical energy model and the proposed recursive architecture will be implemented in a multi-agent simulator to verify its usefulness and analyze the lifetime evaluation of nodes.

## 5.2 Validation of recursive architecture

In this section, the proposed recursive architecture and analytical energy model are implemented in a multi-agent platform (NetLogo) to build a virtual communicating concrete. The NetLogo software environment is authored by Uri Wilensky. It allows to construct and explore agent-based models. This tool has been widely used in many fields of research, especially for the simulation of natural and social phenomena.

A CM energy simulator has been developed for the implementation of the proposed analytical energy models and recursive architecture, whose user interface is shown in Figure 84. It allows defining experimental parameters (network structure, energy consumption of battery, power and communication parameters of communicating module and so on...); implementing different data collection strategies (gathering without data aggregation, with lossless aggregation, lossy aggregation, or some other gathering strategies); analyzing statistical results (first node death time, or active duration of nodes during different phases, etc). Based on the defined parameters and strategies, this simulator allows to visualize the digital part of the WSN in a 3D view as shown in Figure 85, where two communicating elements (in green and yellow)

and their respective communicating nodes are represented as agents. In addition, the energy of nodes is represented by different colors.

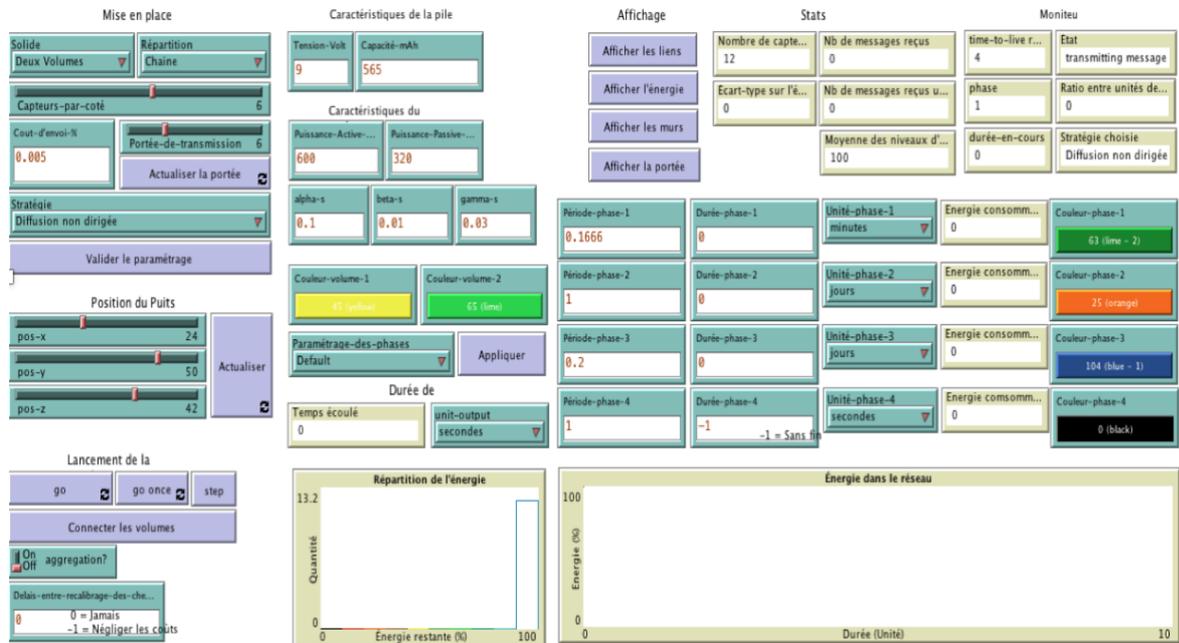


Figure 84 NetLogo McBIM energy simulator user Interface

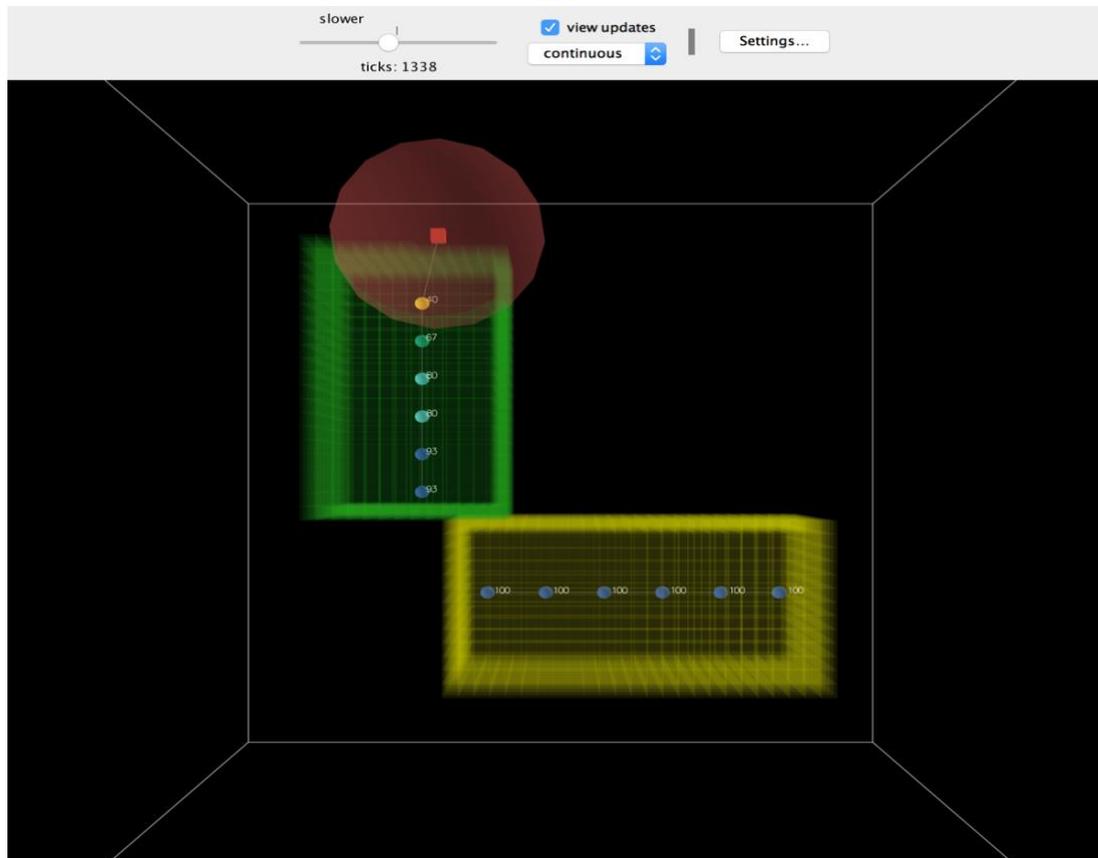


Figure 85 NetLogo McBIM energy simulator 3D view

Agents are used to represent the communicating nodes. Based on the measured parameters with the tests of physical platform, the simulator helps to visualize the energy evolution of the network. The proposed recursive architecture and analytical energy consumption method are implemented in each agent. Moreover, this simulator can be used to analyze energy evolution of nodes for different cooperation methods. To validate the usefulness of this virtual twin of CM, the generated relationship matrix, aggregation coefficients and active duration, generated automatically, are compared with the theoretical results (*section 5.2.1*). Second, the cooperation of different communicating concretes is analyzed in *section 5.2.2*.

### 5.2.1 Analysis of energy consumption within a single communicating concrete

To validate the correctness of the proposed energy consumption methods for every kind of WSN structures, some chain, cluster and tree-based structures (or any combination) are tested. An example of these three kinds of structures is shown in Figure 86. The element 1 is a 5-node chain structure where all nodes (with round shapes) send data to next until the chain leader (node 5 with a square shape). The chain leader then transmits data to the sink node (in red color). The element 2 is a tree structure, where child nodes send data to parent node until node 4 which transmits data to the sink node. The element 3 is a cluster structure where node 5 collects information from members and then transmits information to the sink node.

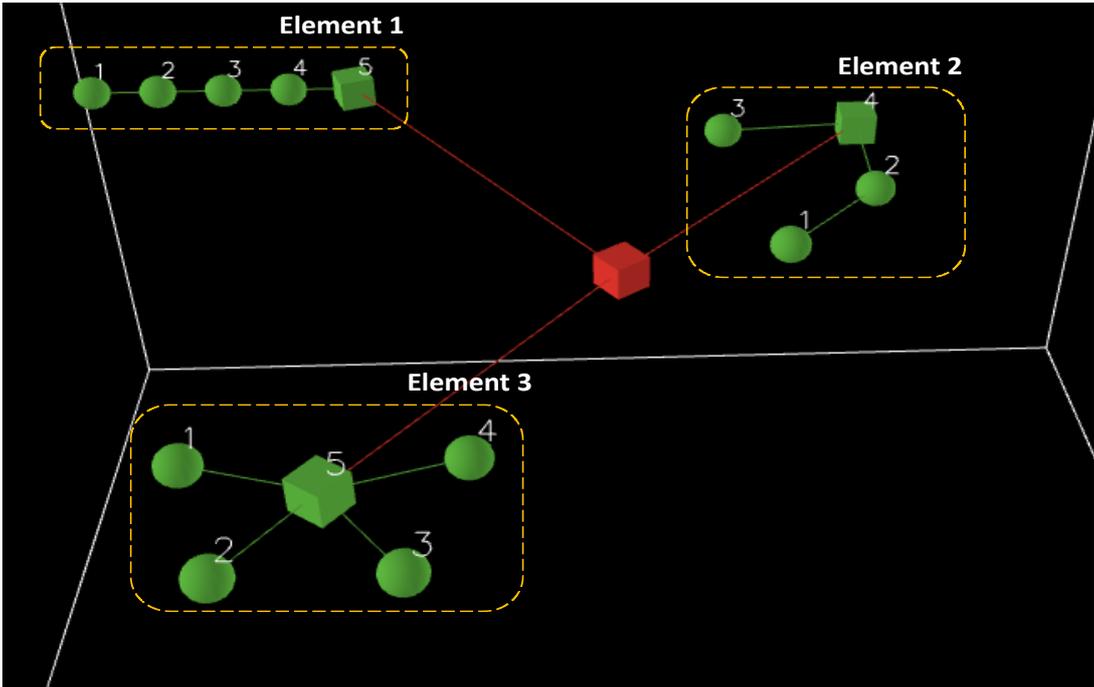


Figure 86 Different structures in NetLogo MAS simulator

As mentioned in *chapter 3*, the proposed analytical model can be used for any type of WSN structures. To compute energy consumption in a structure, it needs to calculate the relationship matrix  $P$ , the gathering matrix  $G$ , the size of received messages  $M_{rx}$  and transmission messages  $M_{tx}$  as well as the active duration. Therefore, to validate the correctness of proposed method, the obtained results by our NetLogo MAS simulator will be compared with theoretical results. Based on the network structure, the relationship matrix  $P$  for these three elements can be computed as in (50), (51) and (52). The gathering matrix depends on the relationship matrix and on the aggregation strategy. In order to simplify the computation, the matrices for collection with data lossless aggregation are compared in this study. The gathering matrices  $G$  for the three structures can then be computed in (53), (54) and (55) for lossy aggregation. Based on the matrix  $P$  and  $G$ , the size of transmitted message and received message can be computed to analyze the active duration and so the energy consumption of every node.

$$P_{element\_1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (50)$$

$$P_{element\_2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad (51)$$

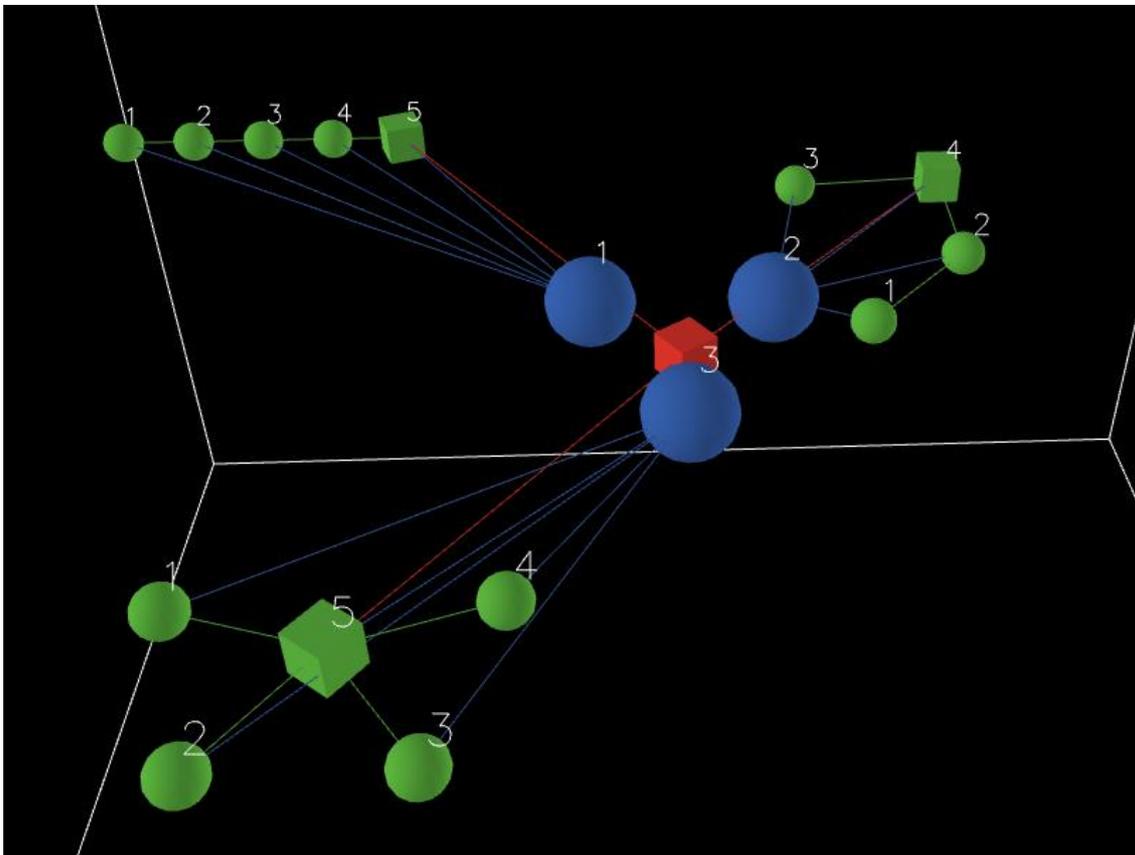
$$P_{element\_3} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (52)$$

$$G_{element\_1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (53)$$

$$G_{element\_2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (54)$$

$$G_{element\_3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (55)$$

In the simulation, each node is represented by an agent which can only know its neighbors. Aggregated agents (in blue color) will be created when the composition process (see Figure 87) is completed. The aggregated agents have the global view (the relationship matrix  $P$ , the gathering matrix  $G$  and energy information) of their related network. Based on the network information, these aggregated agents can then compute the size of received messages and transferred messages and active duration for all members.



*Figure 87 Virtual network and multi-agent architecture in NetLogo*

The matrices obtained by the aggregated agents in NetLogo are presented in Figure 88. The relationship matrices  $P$  and gathering matrices  $G$  in chain, tree and cluster structure are the same as the theoretical matrices in (50) to (55). Based on these matrices, the aggregated agents can also compute the size of the transmitted and received messages, the active duration as well as the energy consumption of all nodes.

Relationship matrix P {{{matrix: [[0 0 0 0][1 0 0 0][0 1 0 0][0 0 1 0][0 0 0 1]]}}}	Relationship matrix P {{{matrix: [[0 0 0 0][1 0 0 0][0 0 0 0][0 1 1 0]]}}}	Relationship matrix P {{{matrix: [[0 0 0 0][0 0 0 0][0 0 0 0][0 0 0 0][1 1 1 1]]}}}
Aggregation coefficient matrix lambda {{{matrix: [[1 0 0 0][1 1 0 0][0 1 1 0][0 0 1 1]]}}}	Aggregation coefficient matrix lambda {{{matrix: [[1 0 0 0][1 1 0 0][0 0 1 0][0 1 1 1]]}}}	Aggregation coefficient matrix lambda {{{matrix: [[1 0 0 0][0 1 0 0][0 0 1 0][0 0 0 1][1 1 1 1]]}}}
Gathering matrix G {{{matrix: [[1 0 0 0][1 1 0 0][1 1 1 0][1 1 1 1]]}}}	Gathering matrix G {{{matrix: [[1 0 0 0][1 1 0 0][0 0 1 0][1 1 1 1]]}}}	Gathering matrix G {{{matrix: [[1 0 0 0][0 1 0 0][0 0 1 0][0 0 0 1][1 1 1 1]]}}}
Data transmission matrix Mtx {{{matrix: [[40][80][120][160][200]]}}}	Data transmission matrix Mtx {{{matrix: [[40][80][40][160]]}}}	Data transmission matrix Mtx {{{matrix: [[40][40][40][40][200]]}}}
Data reception matrix Mrx {{{matrix: [[0][40][80][120][160]]}}}	Data reception matrix Mrx {{{matrix: [[0][40][0][120]]}}}	Data reception matrix Mrx {{{matrix: [[0][0][0][0][160]]}}}
Duration of transmission {{{matrix: [[20][40][70][80][100]]}}}	Duration of transmission {{{matrix: [[20][40][20][80]]}}}	Duration of transmission {{{matrix: [[20][20][20][20][100]]}}}
Duration of reception {{{matrix: [[0][50][70][100][110]]}}}	Duration of reception {{{matrix: [[0][50][0][120]]}}}	Duration of reception {{{matrix: [[0][0][0][0][200]]}}}
a. 5-node chain structure (element 1)	b. tree structure (element 2)	c. cluster structure (element 3)

Figure 88 Energy consumption information obtained by NetLogo

## 5.2.2 Using the simulator to analyze the First Node Death time

Network is damaged if one node dies, therefore, analyzing the first node death (FND) time is crucial. In these three kinds of structure, a special leader node (chain leader, cluster leader or the tree root node), which collects network information from members and transfers information to the sink, always consumes much more energy. As a consequence, this node is the first to die in a classic WSN without any reorganization phase, leading to the end of the data collection. Studying the remaining energy of this node is thus very relevant to maximize the network lifetime. In the following, the developed energy simulator is used to study the FND time of the leader node, depending on the chosen network structure. The active duration of the leader node for both data collection with lossless aggregation and with lossy aggregation are shown in Figure 89 and Figure 90 respectively.

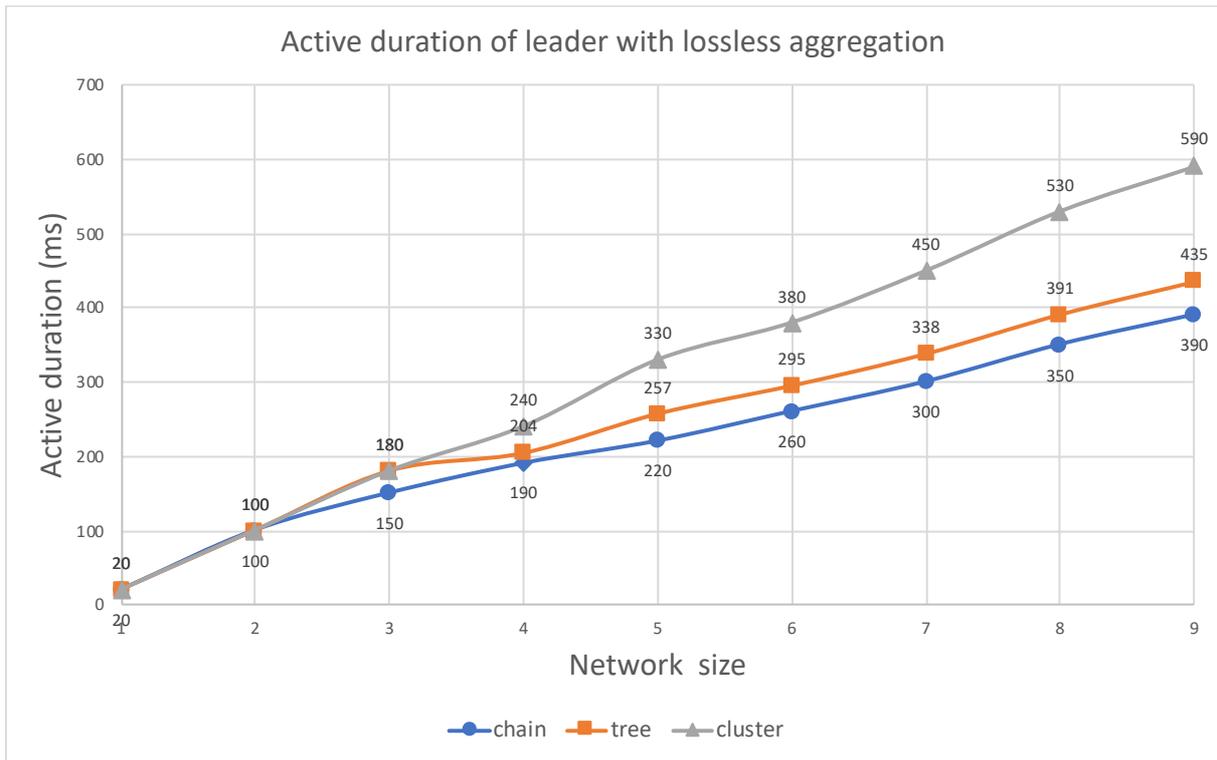


Figure 89 Active duration of leader for data collection with lossless aggregation

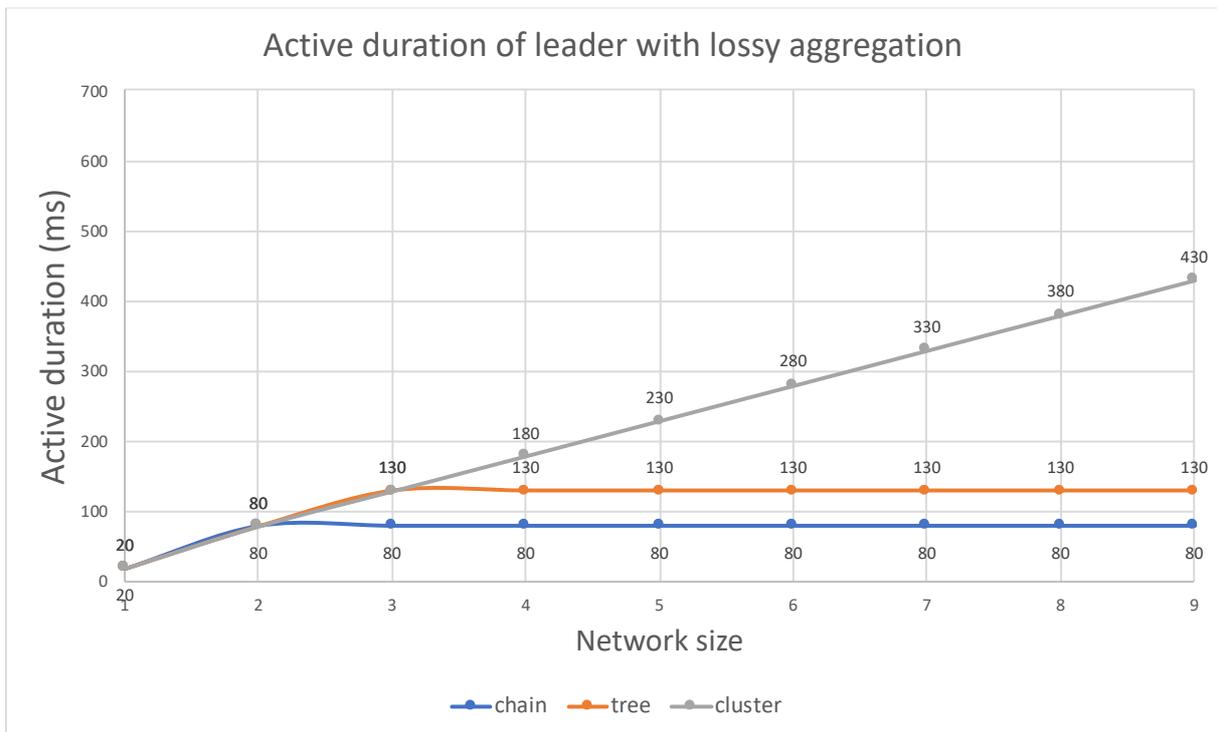


Figure 90 Active duration of leader in data collection with lossy aggregation

For data collection with lossless aggregation, the greater the number of communicating nodes, the longer the active duration at leader is. Compared with tree and chain, the leader node in a cluster is more awake, because it has more children to handle. The more children, the longer

the active duration. Compared with lossless aggregation, the reduction of active duration for the leader in chain and tree for lossy data aggregation collection is obvious. As the child node number does not increase these two structures (from 2 nodes to 9 nodes), active duration of the chain leader and tree root does not increase with the increasing of network size. However, the cluster head has more and more children with the expansion of the network, explaining why the active duration of the cluster head still increases linearly for lossy data aggregation data collection.

The results of FND time with lossy aggregation for the three structures are shown in Figure 91 where each structure is composed of 9 nodes. Each node is powered with a 9 v, 565 mah battery. The longer the active duration per cycle, the shorter the FND delay. As a cluster head has more radio activity per cycle than a chain leader or a root node in a tree, it consumes more energy than the others and reaches FND time first. As a root node in a tree has more children than a chain leader in a single chain structure, it takes more time for replying ACK and thus the tree root node has less lifetime.

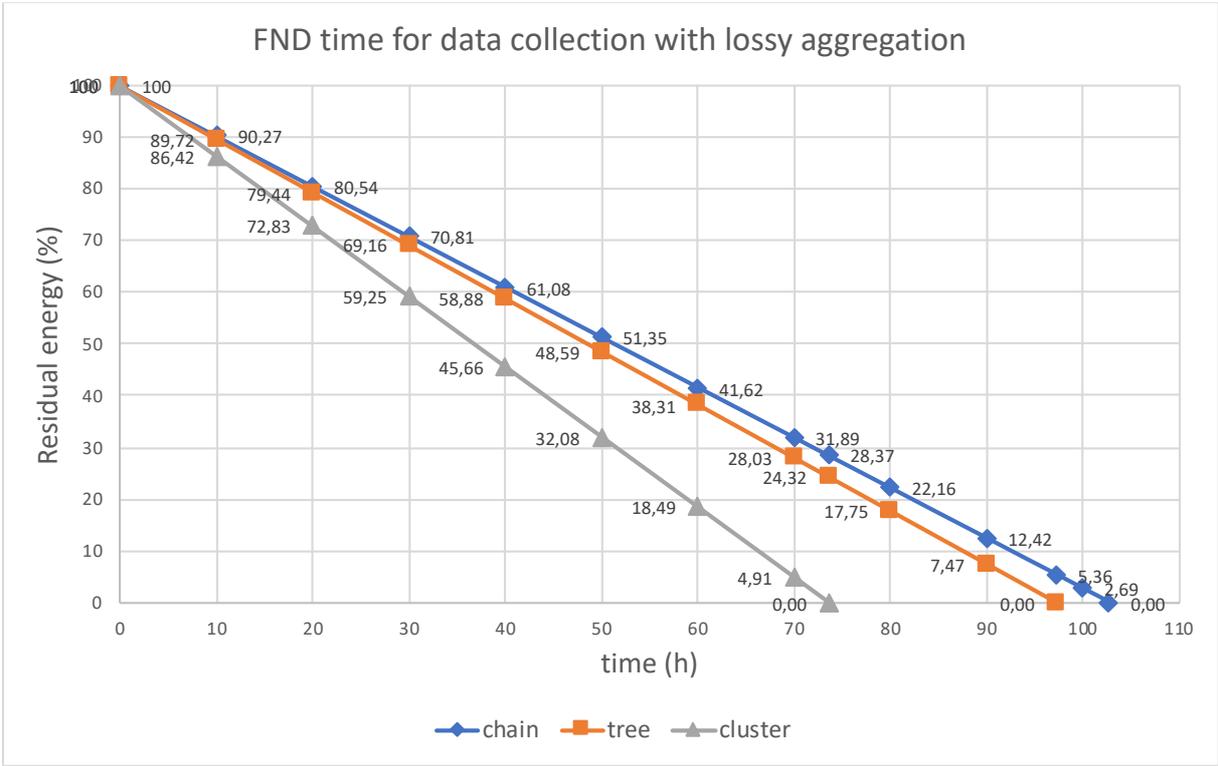


Figure 91 FND time for data collection with lossy aggregation

Besides, aggregated agents in both structures can provide and manage the mean, maximal or minimal energy information which can be used to analyze the network lifetime. For example, we can compare the average residual energy of nodes for the case of first node death. The FND

times of these three structures are 102,77 h, 97,26 h and 73,61 h respectively. Although the FND time in cluster and tree structure is shorter than that in chain, the average residual energy in chain, tree and cluster are 0,73%, 5,7% and 28,84% respectively. Based on the residual energy information, different reconstruction strategies can then be analyzed to adjust the network structure. Besides, the MAS simulator can also be used to predict the consumed energy for different cooperation methods between communicating elements, this usage is detailed in next section.

### 5.2.3 Cooperation of communicating concretes

Communicating elements may cooperate with each other to extend lifetime or complete some special tasks. Energy consumption for different cooperation methods (associated or aggregation) can then be analyzed for the previous two independent communicating elements: a 4-node tree in element 1, and 5-node cluster in element 2 (see Figure 61). The digital model for those two elements is implemented as shown in Figure 92.

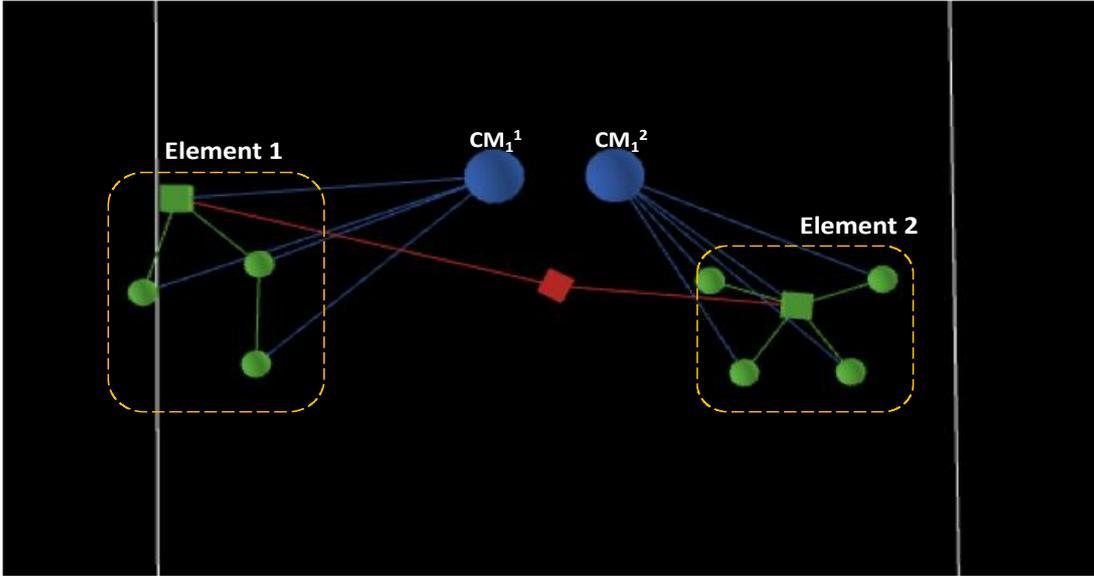


Figure 92 Two independent communicating concretes

The tree structure is on the left, the cluster structure is on the right side of the sink. Agents in green color (level 0) represent the communicating nodes, agents in blue color represent the aggregated agents in level 1. These two structures will both move towards the sink. The composition process will be triggered if one agent detects some agents in the other structure.

The cooperation process for this example is illustrated in Figure 93. At the beginning, the agent  $CM_0^6$  detects the agent  $CM_0^1$  in another element, this agent  $CM_0^6$  then sends a *warning* message to its parent agent. In this case, the agent  $CM_1^2$  sends a *request\_coop* message including its network relationship matrix  $P$ , aggregation matrix  $\lambda$  to the parent agent  $CM_1^1$  using the *PO* operator via the agents  $CM_0^1$  and  $CM_0^6$  as shown in Figure 94. Based on the received information, the agent  $CM_1^1$  will reply with an *ACK\_coop* message which includes the cooperation method (independent, association or aggregation).

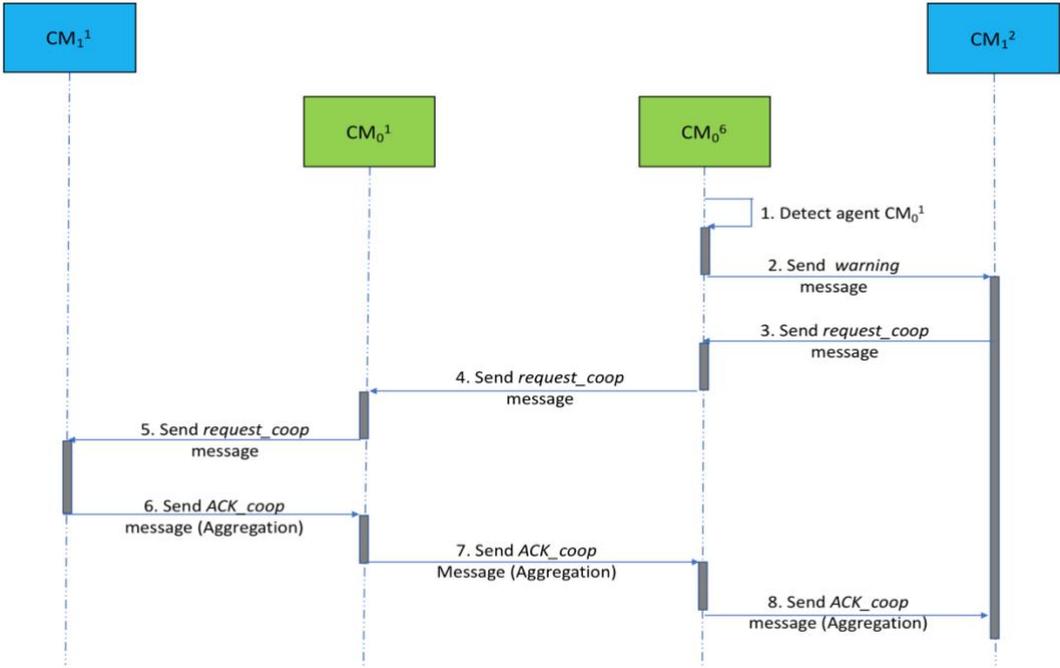


Figure 93 Cooperation between two communicating material elements

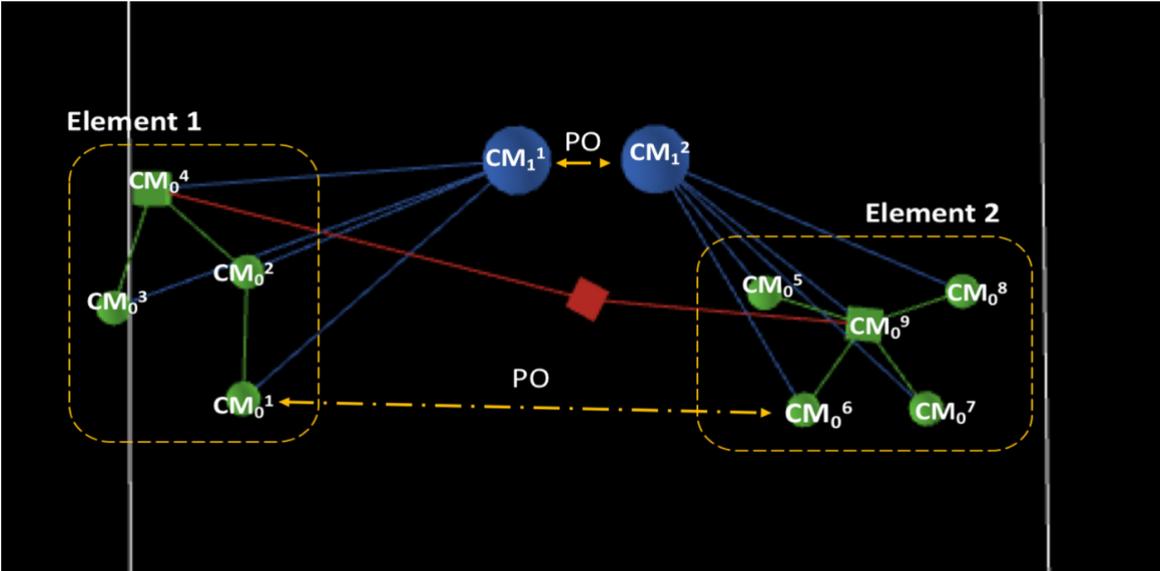


Figure 94 Cooperation between the aggregated agents

The composition process (*PA* operator) will be triggered to create the agent in *level 2* if the chosen cooperation method is *aggregation*. Once the composition process is completed, the generated agent  $CM_2^1$  has a global view of the two CM elements as shown in Figure 95. It can analyze the network lifetime for different routing strategies to extend lifetime. For example, the nodes in the element 1 can transmit data to the BS via the chain leader of the element 2 instead of transmitting directly to the sink as shown in Figure 96.

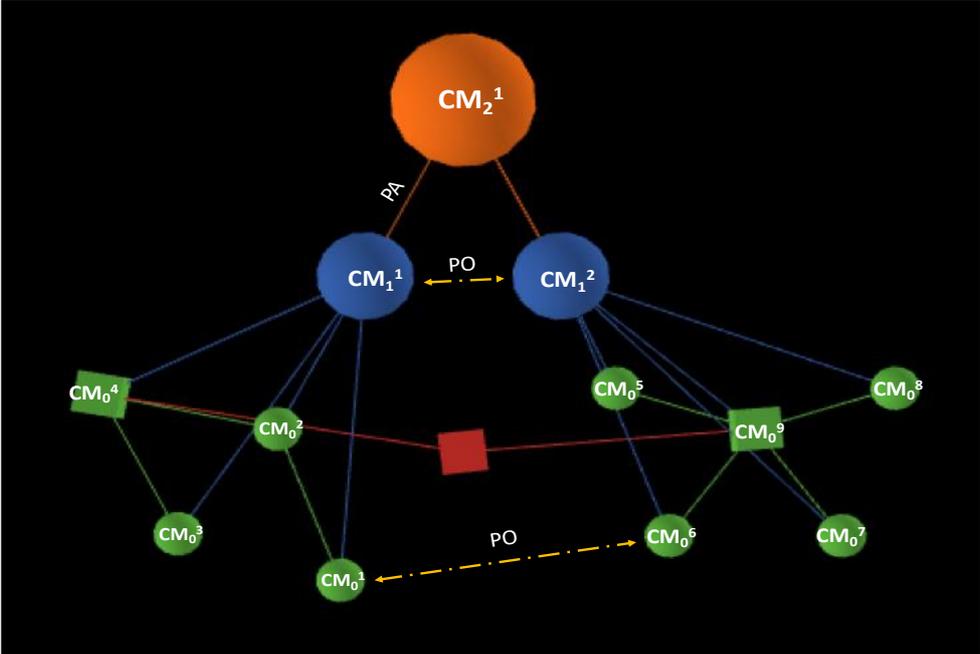


Figure 95 Composition of two communicating concrete

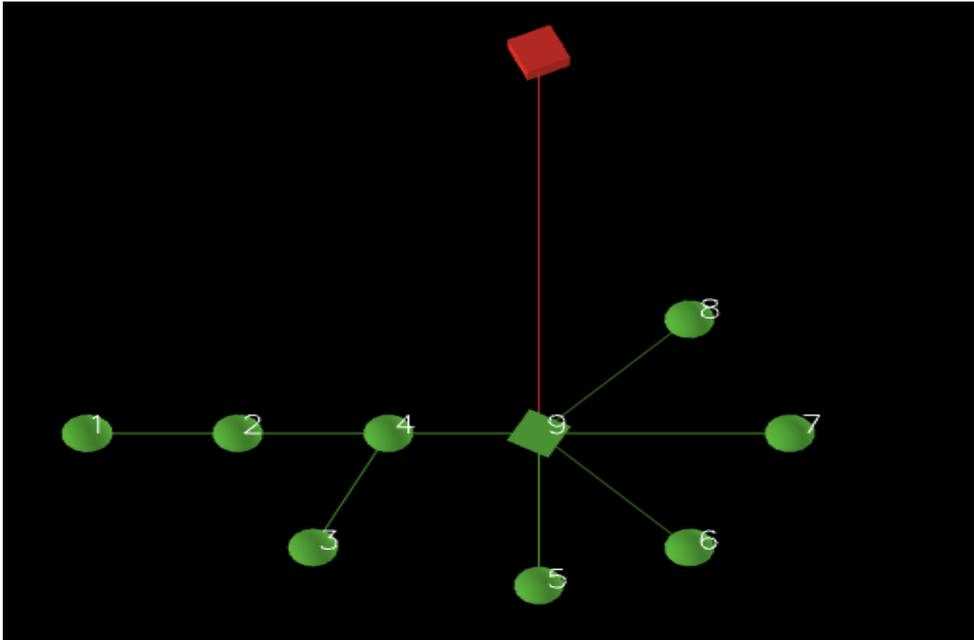


Figure 96 Reconstructed network

Therefore, the new relationship matrix  $P_2^1$  at the aggregated agent  $CM_2^1$  can be computed as in (56). Based on the relationship matrix, the gathering matrix  $G_2^1$  can be calculated as in (57).

$$P_2^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (56)$$

$$G_2^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (57)$$

Based on these two matrices, the active duration of every node can be computed for both lossless and lossy data aggregation collection as in *section 4.6*. For each round of collection, energy consumption at nodes is illustrated in Figure 97. Energy consumption for the end nodes (node 1, 3, 5, 6, 7, 8) is the same for both lossless and lossy strategies, because they don't have child node. For the others, the more child nodes, the more energy it consumes.

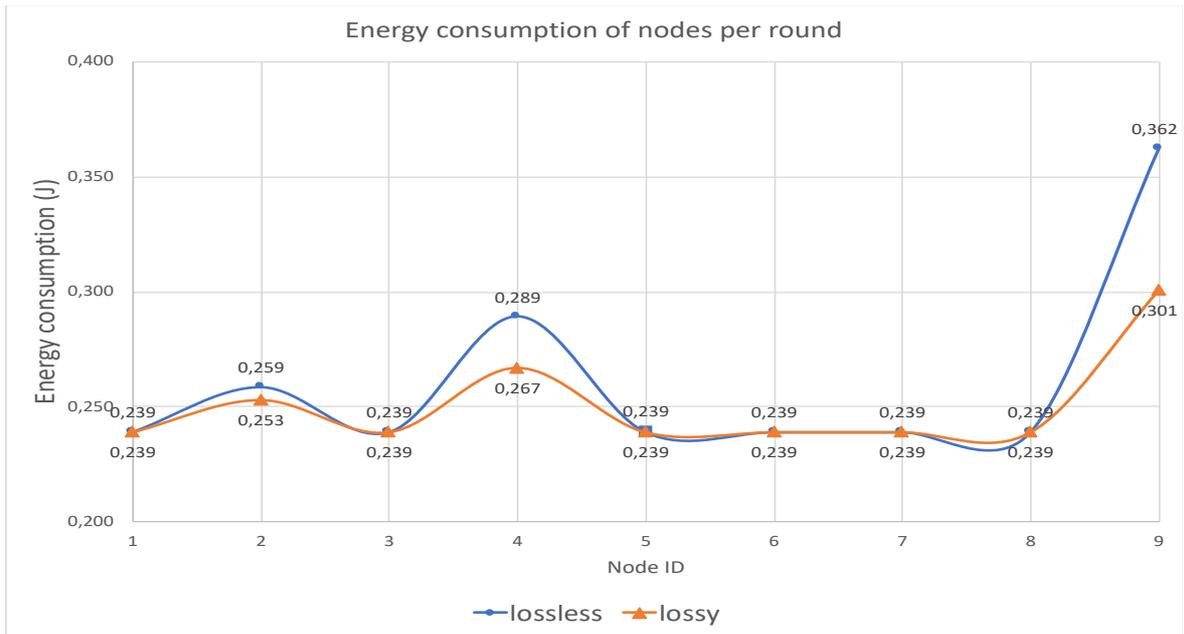


Figure 97 Energy consumption of nodes per round

With the energy consumption of nodes, their lifetime can then be analyzed for both lossless aggregation and lossy aggregation data collection strategies. Since energy consumption depends on the number of child nodes, the lifetime of node 1, 3, 5, 6, 7 and 8 will be the same with a weak consumption. Therefore, only the nodes 1, 2, 4, 9 are considered for lifetime analysis. The results for data collection with lossless aggregation and lossy aggregation are shown in Figure 98 and Figure 99 respectively.

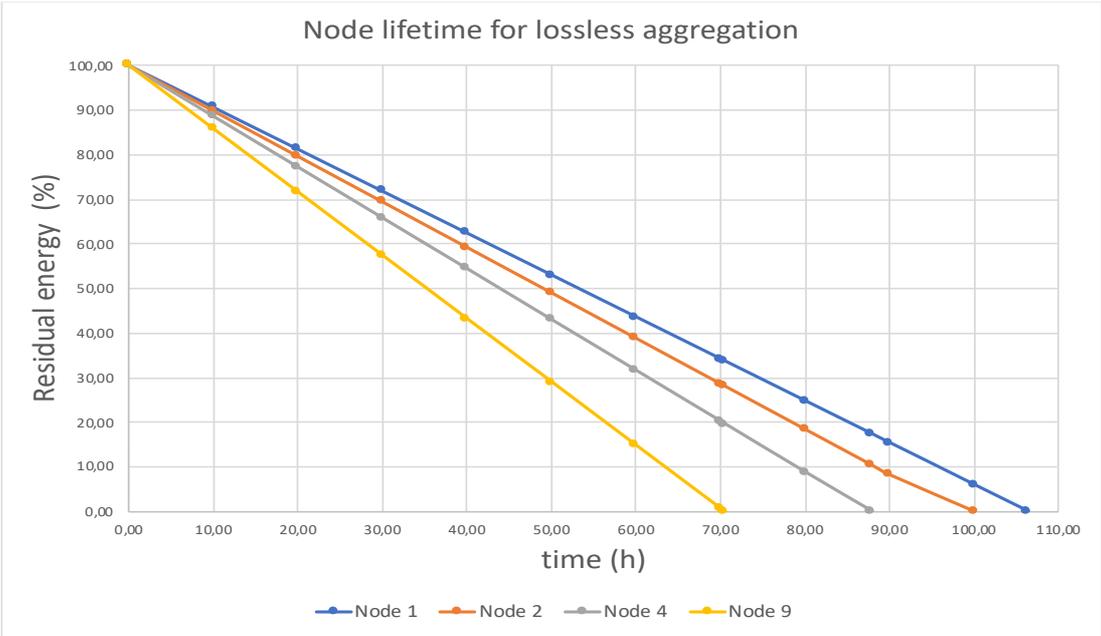


Figure 98 Node lifetime for lossless aggregation

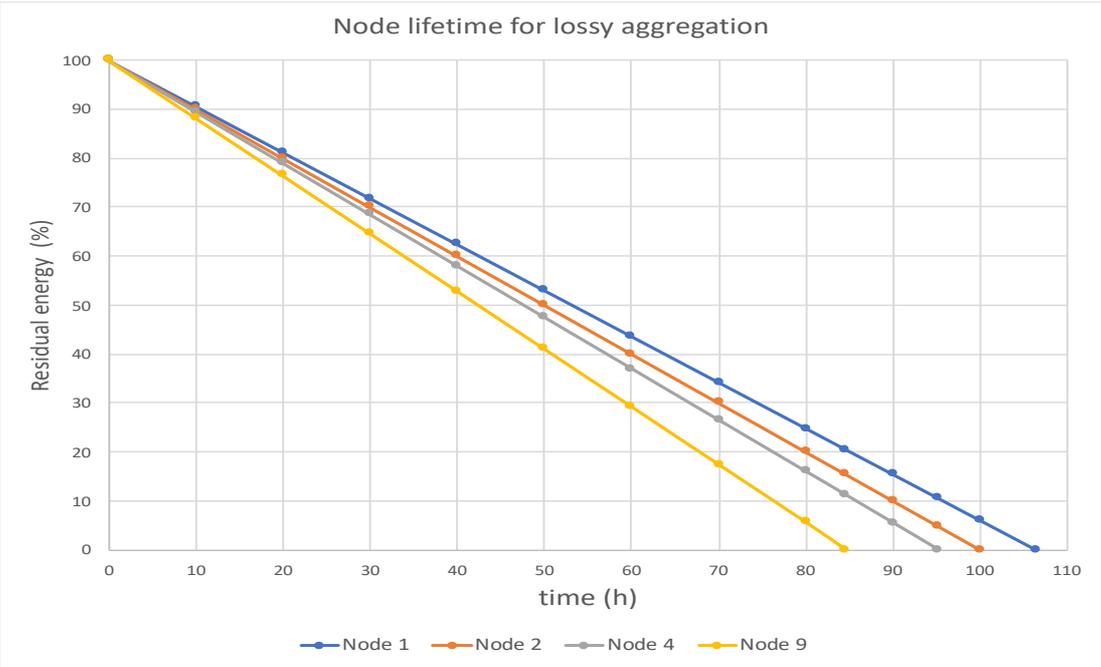


Figure 99 Node lifetime for lossy aggregation

From these results, it is obvious that node lifetime is extended with lossy data aggregation method. The lifetime gain for node 1, node 2, node 4 and node 9 are 0%, 2,24%, 8,33% and 20,49% respectively. The greater the number of children, the more obvious the extended life.

### 5.3 Analysis

This chapter presents experimental results for both the accuracy of the proposed energy model in physical part and the usefulness of the recursive architecture for energy estimation in digital part. In the physical part, the proposed analytical estimation models were verified and validated by a chain experiment platform. Using sleep mode combined with data aggregation greatly reduces energy consumption. According to the node position in the chain, the reduction can be quantified by the gap of active duration between the two collection modes (with or without data aggregation). For a 6-node chain, the closer to the chain leader, the more obvious the reduction in energy consumption. For one round, the active duration of chain leader for data collection without aggregation is 3,5 times longer than in the case of aggregation. The proposed analytical energy estimation model achieves a high accuracy (more than 95%) for ten iterations. The long cycles test will be performed in following works, such as one week or one month.

In different phases, communicating concretes may cooperate with each other to construct a new network structure. In the proposed recursive architecture, three operators ( $PA$ ,  $PO$  and  $PI$ ) are detailed for the composition, cooperation process and message exchanges between agents. Energy estimation models are implemented in a 3D multi-agent simulator NetLogo for analyzing energy evolution in different cooperation possibilities. Aggregated agents are created automatically when the composition conditions are met. They have the global view of their members (the relationship matrix  $P$ , the gathering matrix  $G$  and energy information), and can then compute the size of received and transferred messages as well as active duration of all members. Based on the computed energy consumption information, the aggregated agent can provide us the remaining energy information (such as the FND time) which can be used to analyze different data collection strategies to extend lifetime in a resource constrained application like the McBIM project.



# General conclusion and future works

This thesis presents the communicating material paradigm as well as a holonic architecture for data and energy management. The study case is the McBIM project where concretes are equipped with embedded WSN. In the proposed architecture, the communicating material is represented by a virtual agent in digital part, which can store product-related information and provide optimal solutions to meet environment changes and extend the lifetime. As the communicating concrete should work ten or twenty years, energy management is relevant for this resource constrained application. As consequence, this work proposes analytical energy estimation models for periodical data collection with WSN to provide a rapid and approximated energy information. Besides, communicating concrete may cooperate with each other to build larger network and save energy. As a result, a recursive MAS architecture is proposed to efficiently manage informational part (the agents in digital part). As the analytical energy model is implemented in the digital part, energy efficient strategies may be provided with a global view and an optimal solution to extend product lifetime.

The *chapter 1* reviews the use of intelligent product in product lifecycle management and introduces the context and the concept of communicating material. Based on the CM concept, recent works developed at the CRAN are presented as well as their interests for PLM. Although the concept of CM has been studied for a long time, no clear formal definitions of the communicating material have been provided until now. As consequence, a definition is provided underlying the fact that CM is a recursive concept (since a CM can be composed of other CMs). As this thesis is in the framework of McBIM project which aims to apply this concept to industrial concrete, the communicating concrete design is presented as well as its position in CM paradigm. In order to realize the communicating concrete design, and make it work as long as possible, new research issues in the physical part (data collection in WSN) and in the digital part (energy and information management in MAS) are underlined. For the physical part, the research issues are **how to maximize lifetime of data collection within the embedded WSN and predict remaining energy of network with a simple analytical energy estimation model?** And for the digital part, the research issues are: **how to manage monitoring data and energy information at agents? how to organize the agents to achieve a recursive architecture and how to realize the composition?**

To address these research issues, states of the art regarding the physical part and digital part are described in the *chapter 2*. In the physical part, the existing data aggregation solutions, routing protocol and time synchronization protocols are analyzed to find an efficient way for periodical data collection. Using data reduction techniques at nodes allows to reduce transmission messages and extend network lifetime. The time synchronization protocols always depend on the applied routing protocols. Each data collection structure has their special benefits, it is thus hard to ensure that a routing protocol can be suitable for all lifecycle phases. Most of works apply the distance-based energy models. However, distance seems to be unsuitable for WSNs in our case study where obtaining the node positions is unrealistic. Therefore, a general analytical energy model, using a power-based consumption approach, is needed for periodical data collection in a network of communicating concretes.

In the digital part, the existing recursive holarchies and MAS architecture are studied to derive a recursive architecture for CM applications. The needs of recursion are in different applicative areas, and most of proposed architectures are for static environments. However, communicating concrete may aggregate with others throughout its whole lifetime. A dynamic architecture is then required. Although the application for WSN in Ocelllo is closest to our needs, more information should be added to the agent model, and the network cooperation between WSNs is not considered. Thus, a dynamic recursive architecture is needed for a CM application, as well as a formalization of the composition and cooperation processes for dynamic scenarios.

Regarding the analytical energy estimation model, the *chapter 3* presents a power-based energy consumption model for communicating node. As energy consumption depends on the duration of active and sleep activities, the duration is analyzed for different cases. In the first part, an analytical energy model based on the position of nodes in chain structure has been proposed to estimate energy consumption. Data aggregation and time synchronization are both considered to reduce energy consumption at nodes. To provide approximate remaining energy of nodes for all types of structures, a relationship-based analytical energy model has then been proposed and detailed. For a given data collection structure where aggregation strategies are defined by users, the size of transmitted and received messages can be computed with the defined algorithm which can then be used to calculate energy consumption at nodes to estimate their residual lifetime. This energy consumption information of nodes in a resource constrained environment can not only provide user the remaining lifetime but also can help to select appropriate data collection strategies to meet the environment's changes.

To efficiently manage the informational part of the CM, a recursive architecture has been proposed in *chapter 4*. Based on MAS-R, an adapted agent model has been proposed for the communicating concrete application, where communicating relationship matrix, aggregation coefficients as well as other parameters for building the analytical energy model are added. To realize the recursive architecture, the composition/decomposition processes have been formalized as well as the lifetime estimation process to provide the remaining energy information. As a communicating concrete may cooperate with others in some phases of its whole lifetime, different cooperation possibilities have been discussed, and the cooperation process has been explained. With this recursive architecture, aggregated agents will be generated if the composition conditions are met. Once the composition process is completed, the agent in the highest level has the global view of energy information and communication relationship, this agent can then estimate the remaining lifetime and select suitable data collection strategies to extend CM product lifetime.

The *chapter 5* presents the experimental results for both the physical part (energy estimation model for data collection in WSN) and the digital part (CM recursive architecture for energy management). For the physical part, periodic data collection program with/without data aggregation is implemented in an Arduino-XBee platform to verify the accuracy of the proposed analytical energy consumption models. According to the node position in the chain, the reduction can be quantified by the gap of active duration between the two collection modes (with or without aggregation). The results showed that using sleep mode combined with data aggregation greatly reduces energy consumption. The proposed analytical energy estimation model achieves a high accuracy where the average error is around 5%. The duration of radio activity depends on the used communicating technology, the values of the  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $d_{syn}$  parameters ( $\alpha$  for transmission,  $\beta$  for IFS,  $\gamma$  for ACK and  $D_{syn}$  for synchronization) can thus be adapted for any wireless network communication technology. Therefore, this energy estimation model is thought to be generic enough. It has been implemented in the CM agents to analyze their lifetime evolution.

In the second part of the *chapter 5*, the proposed recursive architecture, composition and cooperation processes are implemented in the 3D multi-agent simulator NetLogo. An aggregated agent is created once the composition process is completed in the simulation. The aggregated agent in the highest level has the global view of all members (their communication relationship  $P$ , their gathering matrix  $G$  and their energy information, etc...). In the case of

cooperation between concretes, once an agent finds another available communicating elements, the cooperation process is triggered. An aggregated agent in an upper level is created if the cooperation method is aggregation. This agent in the upper level can compute the active duration of its members, and it can provide the remaining energy information which can be used to analyze different data collection strategies to extend lifetime.

## **Perspectives**

The perspectives in short-term is mainly in the physical part: we will improve, test and adapt energy estimation models to different aggregation routing protocols (tree-based and cluster-based or a hybrid structure). Extended experimentations could be made in order to consolidate models. Three kinds of campaigns can be envisaged: 1. Long time experimentations to verify that the models do not drift from reality with time; 2. Experimentations with a longer chain structure to experimentally confirm the gap between collection with or without aggregation; 3. Experimentations with other wireless technologies to compare performances. In addition, the proposed energy estimation model as well as the recursive architecture will be integrated into the McBIM project, and link to a real communicating concrete. Both the composition and cooperation processes of nodes within a single concrete will be tested as well as the cooperation between several communicating concretes. Lifetime estimation in different scenarios will be analyzed and verified to illustrate its usefulness.

For long-term perspectives, we would like to combine the estimation model with decision-making (reconfiguration of the collection process) through centralized or distributed algorithms (or even a hybridization of the two) and to compare this approach with existing solutions. For example, we can use machine learning or other AI methods to evaluate and select the most suitable and efficient data gathering strategies with the product-related information (life phase, remaining energy and so on) to extend product lifetime. Meanwhile, we will test with other communication technologies to show the genericity of the proposed model. Besides, some other resource constrained applications may be considered for the use of the proposed holonic architecture. For example, the transportation of COVID-19 vaccine, WSN can be used to monitor temperature of the container to ensure the safety of vaccine. In a forest, monitoring temperature and humidity can be used to prevent fires. The proposed holonic architecture will be tested with other intelligent products or communicating materials to demonstrate its usefulness for the product lifecycle management.



# References

- Abdoos, M., Mozayani, N. and Bazzan, A. L. C. (2013)** ‘Holonic multi-agent system for traffic signals control’, *Engineering Applications of Artificial Intelligence*. Elsevier, 26(5–6), pp. 1575–1587.
- Abdulsalam, H. M. and Kamel, L. K. (2010)** ‘W-LEACH: Weighted low energy adaptive clustering hierarchy aggregation algorithm for data streams in wireless sensor networks’, *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 1–8.
- Afroz, F. and Braun, R. (2020)** ‘Energy-efficient MAC protocols for wireless sensor networks: A survey’, *International Journal of Sensor Networks*, vol. 32, no.3, pp. 150–173.
- Ahn, K. S., Kim, D. G., Sim, B. S., Youn, H. Y. and Song, O. (2011)** ‘Balanced chain-based routing protocol(BCBRP) for energy efficient wireless sensor networks’, *Proceedings - 9th IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops*, 2011, pp. 227–231.
- Alfayez, F., Hammoudeh, M. and Abuarqoub, A. (2015)** ‘A Survey on MAC Protocols for Duty-cycled Wireless Sensor Networks’, *Procedia Computer Science*. vol. 73, pp. 482–489.
- ANR Project McBIM (2017)**. Available at: <http://mcbim.cran.univ-lorraine.fr/Members>.
- Aziz, A. , Sekercioglu, Y. A., Fitzpatrick, P.,and Ivanovich, M. (2013)** "A Survey on Distributed Topology Control Techniques for Extending the Lifetime of Battery Powered Wireless Sensor Networks," in *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 121-144.
- Bărbat, B., Căndea, C. and Zamfirescu, C. (2001)** ‘Holons and Agents in Robotic Teams A Synergistic approach’, *In proceedings of ENAIS'2001*, pp. 654 – 660.
- Belle, J.V., Germain, B.S., Valckenaers, P., Brussel, H.V., Bahtiar, R., and Cattryse, D. (2011)** ‘Intelligent products in the supply chain are merging logistic and manufacturing operations’, *IFAC Proceedings Volumes*, vol.44 pp. 1596–1601.
- Bhuiyan, M. Z. A, Wang, G., Cao, J., and Wu, J.(2015)** ‘Deploying Wireless Sensor Networks with Fault-Tolerance for Structural Health Monitoring’, *IEEE Transactions on Computers*, vol. 64, no. 2 pp. 382–395.
- Botti, V. and Boggino. A. G. (2008)** ‘ANEMONA: A Multi-agent Methodology for Holonic Manufacturing Systems’, *Springer Series in Advanced Manufacturing*.
- Braman, A. and Umapathi, G. R. (2014)** ‘A Comparative Study on Advances in LEACH Routing Protocol for Wireless Sensor Networks: A survey’, *International Journal of Advanced Research in Computer and Communication Engineering*, vol.3, no.2, pp. 5683–5690.
- Brussel, V., Wyns, J., Valckenaers, P., Bongaerts, L. andPeeters, P(1998)** ‘Reference Architecture for Holonic Manufacturing Systems’, *Computers in Industry*, vol. 37, no. 2, pp. 255–274.
- Caridi, M. and Cavalieri, S. (2004)** ‘Multi-agent systems in production planning and control: an overview’, *Production Planning & Control*, vol.15, pp. 106–118.
- Chirn, J. L. and McFarlane, D. C. (2000)** ‘A holonic component-based approach to reconfigurable

manufacturing control architecture', *Proceedings - International Workshop on Database and Expert Systems Applications, DEXA*, 2000-Janua, pp. 219–223.

**Cui, J. (2017)** 'Data Aggregation in Wireless Sensor Networks', *Thèse de doctorat, Université de Lyon*.

**Dialog (2003)**. Available at: [http://dialog.hut\\_](http://dialog.hut_).

**Ding, M., Cheng, X. and Xue, G. (2004)** 'Aggregation tree construction in sensor networks', *IEEE 58th VTC*, vol.4, pp. 2168-2172.

**Dorri, A., Kanhere, S. S. and Jurdak, R. (2018)** 'Multi-Agent Systems: A Survey', *IEEE Access*. IEEE, vol. 6, pp. 28573–28593.

**Dron, W., Duquenooy, S., Voigt, T., Hachicha, K. and Garda, P. (2014)** 'An emulation-based method for lifetime estimation of wireless sensor networks', *Proceedings - IEEE International Conference on Distributed Computing in Sensor Systems*, pp. 241–248.

**Du, W., Mieleville, F. and Navarro, D. (2010)** 'Modeling energy consumption of wireless sensor networks by SystemC', *Proceedings - 5th International Conference on Systems and Networks Communications*, pp. 94–98.

**EPCglobal**. Available at: <http://www.epcglobalinc.org/home/>.

**Esmaili, A., Mozayani, N., Motlagh, M. R. and Matson, E. T.(2017)** 'A socially-based distributed self-organizing algorithm for holonic multi-agent systems: Case study in a task environment', *Cognitive Systems Research*. Elsevier, vol. 43, pp. 21–44..

**Elson, J., Girod, L., and Estrin, D. (2002)** 'Fine-Grained Time Synchronization Using Reference Broadcasts', *Proceedings - the 5th Symposium on Operating Systems Design and Implementation*, 2002, pp. 147–163.

**Fasolo, E., Rossi, M., Widler, J. and Zorzi, M. (2007)** 'in-Network Aggregation Techniques for Wireless Sensor Networks: a Survey', *IEEE Wireless Communication*, vol. 14, pp. 70–87.

**Ferry, N. Ducloyer, S., Julien, N. and Jutel, D. (2011)** 'Power/energy estimator for designing WSN nodes with ambient energy harvesting feature', *Eurasip Journal on Embedded Systems*.

**Ganeriwal, S., Kumar, R. and Srivastava, M. B. (2003)** 'Timing-sync Protocol for Sensor Networks Categories and Subject Descriptors', *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 138–149.

**García Ansola, P., Morenas, J., Higuera, A., and Otamendi, J. (2012)** 'Distributed decision support system for airport ground handling management using WSN and MAS', *Engineering Applications of Artificial Intelligence*, vol. 25, no. 3, pp. 544–553..

**Giret, A. and Botti, V. (2003)** 'Towards a recursive agent oriented methodology for large-scale MAS', *Agent-Oriented Software Engineering IV*, pp. 135–161.

**Gupta, G. and Younis, M. (2004)** 'Load-balanced clustering of wireless sensor networks', *Proceedings - IEEE International Conference on Communications*, 2003, vol.3 pp. 1848–1852..

**Han, Z., Wu, J., Zhang, J., Liu, L. and Tian, K. (2014)** 'A general self-organized tree-based energy-balance routing protocol for wireless sensor network', *IEEE Transactions on Nuclear Science*, vol. 61, no. 2, pp. 732–740.

- Hawbani, A., Wang, X., Kuhlani, H., Karmoshi, S., Ghoul, R., Sharabi, Y. and Tirbosh, E. (2018)** ‘Sink-oriented tree based data dissemination protocol for mobile sinks wireless sensor networks’, *Wireless Networks*. Springer, vol.24, no. 7, pp. 2723–2734.
- Heinzelman, W. B., Chandrakasan, A. P. and Balakrishnan, H. (2002)** ‘An application-specific protocol architecture for wireless microsensor networks’, *IEEE Transactions on Wireless Communications*, vol. , no. 4, pp. 660–670.
- Heinzelman, W. R., Chandrakasan, A. and Balakrishnan, H. (2000)** ‘Energy-efficient communication protocol for wireless microsensor networks’, *Proceedings - the 33rd Annual Hawaii International Conference on System Sciences*, 2000, vol. 2, pp. 10.
- Hoang, T. (2012)** ‘Un modèle multi-agent récursif générique pour simplifier la supervision de systèmes complexes artificiels décentralisés’, *Thèse de doctorat, Université de Grenoble*, 2012.
- Howitt, I., Neto, R., Wang, J. and Conrad, J. M. (2005)** ‘Extended energy model for the low rate WPAN’, *Proceedings - 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2005, pp. 315–322.
- Isern, D., Sánchez, D. and Moreno, A. (2011)** ‘The Journal of Systems and Software Organizational structures supported by agent-oriented methodologies’, *The Journal of Systems & Software*. Elsevier, vol. 84, no. 2, pp. 169–184.
- Jalsan, K. E., Flouri, K. and Feltrin, G. (2014)** ‘Energy consumption estimation for wireless sensor network layout optimization’, *Proceedings - 7th International Conference on Ubi-Media Computing and Workshops*, 2014, pp. 238–242.
- Jan, M. A., Nanda, P. and He, X. (2013)** ‘Energy evaluation model for an improved centralized clustering hierarchical algorithm in WSN’, *Proceedings - International Conference on Wired/Wireless Internet Communication*, 2013, pp. 154-167.
- Jover, J., Thomas, A., Leban, J. M. and Canet, D. (2013)** ‘Interest of new communicating material paradigm: An attempt in wood industry’, *Journal of Physics: Conference Series*, vol. 416, pp. 12031.
- Jun, H. B., Kiritsis, D. and Xirouchakis, P. (2007)** ‘Research issues on closed-loop PLM’, *Computers in Industry*, vol. 58, pp. 855–868.
- Kalpakis, K. and Tang, S. (2009)** ‘A combinatorial algorithm for the maximum lifetime data gathering with aggregation problem in sensor networks’, *Computer Communications*. Elsevier B.V., vol. 32, no. 15, pp. 1655–1665.
- Kärkkäinen, M., Ala-Risku, T. and Främling, K. (2003)** ‘The product centric approach: A solution to supply network information management problems?’, *Computers in Industry*, vol. 52, no. 2, pp. 147–159.
- Kaur, P., Sohi, B. S. and Singh, P. (2019)** ‘Recent Advances in MAC Protocols for the Energy Harvesting Based WSN: A Comprehensive Review’, *Wireless Personal Communications*. Springer, vol. 104, no. 1, pp. 423–440.
- Khedr, A. M., Aziz, A. and Osamy, W. (2021)** ‘Successors of PEGASIS protocol: A comprehensive survey’, *Computer Science Review*. Elsevier, vol. 39, pp. 100368.
- Khelifa, B. and Laouar, M. R. (2020)** ‘A holonic intelligent decision support system for urban project planning

- by ant colony optimization algorithm', *Applied Soft Computing Journal*. Elsevier, vol. 96, pp. 106621.
- Kiritsis, D. (2011)** 'Closed-loop PLM for intelligent products in the era of the Internet of things', *CAD Computer Aided Design*. Elsevier, vol. 43, no. 5, pp. 479–501.
- Koestler, A. (1968)** 'The ghost in the machine'.
- Krommenacker, N., Charpentier, P. and Pena, R. (2013)** 'Towards Product Health Monitoring throughout its life-cycle', *Proceedings – 11<sup>th</sup> 11th IFAC Workshop on Intelligent Manufacturing System*, 2013, pp.169-173.
- Kubler, S., Derigent, W., Thomas, A. and Rondeau, E. (2010)** 'Problem definition methodology for the “communicating material” paradigm', *IFAC Proceedings Volumes*, vol.43, pp. 198–203.
- Kubler, S. (2012)** 'Premiers travaux relatifs au concept de matière communicante : Processus de dissémination des informations relatives au produit', *Thèse de doctorat, Université de Lorraine*, 2012.
- Pal, P., and Chatterjee, P. (2014)** 'A Survey on TDMA-based MAC Protocols for Wireless Sensor Network', *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, pp. 12.
- Lamonaca, F., Sciammarella, P.F., Scuro, C., Carnì, D. L. and Olivito, R. S. (2018)** 'Internet of Things for Structural Health Monitoring', *Proceedings – 2018 Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT*, 2018, pp. 95–100.
- Leitão, P. and Karnouskos, S. (2015)** *Industrial Agents: Emerging Applications of Software Agents in Industry*.
- Leitão, P. and Restivo, F. (2006)** 'ADACOR: A holonic architecture for agile and adaptive manufacturing control', *Computers in Industry*, vol. 57, no. 2, pp. 121–130.
- Li, S., Cui, J. and Li, Z. (2011)** 'Wireless sensor network for precise agriculture monitoring', *Proceedings - 4th International Conference on Intelligent Computation Technology and Automation, ICICTA 2011*, vol. 1, pp. 307–310.
- Lindsey, S., Raghavendra, C. and Sivalingam, K. M. (2002)** 'Data gathering algorithms in sensor networks using energy metrics', *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924–935.
- Loscrì, V., Morabito, G. and Marano, S. (2006)** 'A two-levels hierarchy for low-energy adaptive clustering hierarchy (TL-LEACH)', *IEEE Vehicular Technology Conference*, vol. 3, pp. 1809–1813.
- Loubet, G., Takacs, A., Gardner, E., Luca, A. D., Udrea, F. and Dragomirescu, D. (2019)** 'LoRaWAN battery-free wireless sensors network designed for structural health monitoring in the construction domain', *Sensors*, vol. 19, no. 7. pp. 1510.
- Lu, J., Valois, F., Dohler, M., and Wu, M. Y. (2010)** 'Optimized data aggregation in WSNs using adaptive ARMA', *Proceedings - fourth International Conference on Sensor Technologies*, 2010, pp. 115-120.
- Lu, Y., Comsa, I. S., Kuonen, P. and Hirsbrunner, B. (2015)** 'Dynamic data aggregation protocol based on multiple objective tree in Wireless Sensor Networks', *Proceedings - IEEE 10th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2015, pp. 7–9.
- Ma, L. and Zhang, Y. (2014)** 'Hierarchical social network analysis using multi-agent systems: A school system case', *Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 2014, pp. 1412–1419.

- Ma, Z., Li, G., and Gong, Q. (2009)** ‘Improvement on LEACH-C Protocol of Wireless Sensor Network (LEACH-CC)’, *International Journal of Future Generation Communication and Networking*, vol. 9, no. 2, pp. 183–192.
- Madden, S., Franklin, M. J., Hellerstein, J. and Hong, W. (2002)** ‘TAG: A Tiny AGgregation Service for Ad-hoc Sensor Networks’, *OsdI*, 36(SI), pp. 131–146.
- Mahdi, H. S., Mohammed, I. J. and Baithoon, N. Y. (2015)** ‘Low Transmission Delay Based on LEACH and PEGASIS Protocol in WSN’, *International Journal of Computer Science and Mobile Computing*, vol. 4, pp. 209–217.
- Manisterski, E., Sarne, D. and Kraus, S. (2007)** ‘Enhancing MAS cooperative search through coalition partitioning’, *Proceedings - International Joint Conference on Artificial Intelligence*, 2007, pp. 1415–1421.
- Margi, C. B. and Obraczka, K. (2004)** ‘Instrumenting network simulators for evaluating energy consumption in power-aware Ad-hoc network protocols’, *Proceedings - IEEE Computer Society’s Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, 2004, pp. 337–346.
- Marhoon, H. A., Alubady, R. and Abdulhameed, M. K. (2020)** ‘Direct line routing protocol to reduce delay for chain based technique in wireless sensor network’, *Karbala International Journal of Modern Science*, vol. 6, no. 2, pp. 190–195.
- McFarlane, D., Sarma, S., Chirn, J. L., Wong, C. Y., and Ashton, K. (2002)** ‘The intelligent product in manufacturing control and management’, *IFAC Proceedings*, vol. 35, pp. 49-54.
- McFarlane, D., Sarma, S., Chirn, J. L., Wong, C. Y. and Ashton, K. (2003)** ‘Auto ID systems and intelligent manufacturing control’, *Engineering Applications of Artificial Intelligence*, vol.16, no. 4, pp. 365–376.
- Mekki, K., Derigent, W., Zouinkhi, A., Rondeau, E. and Abdelkrim, M. N. (2014)** ‘Data Dissemination Algorithms for Communicating Materials Using Wireless Sensor Networks’, *Proceedings - International Conference on Future Internet of Things and Cloud*, 2014, pp. 230–237.
- Mekki, K. (2016)** ‘Gestion de l’information embarquée dans des matériaux communicants à l’aide de protocoles de réseaux de capteurs’, *Thèse de doctorat, Université de Lorraine*, 2016.
- Meszyński, S. and Sokolov, O. (2017)** ‘Modeling of Complex Multiagent Behavior Using Matrix Representation Modeling of Complex Multiagent Behavior Using Matrix Representation’, *Proceedings - The Sixth International Conference on Intelligent Systems and Applications*, 2017.
- Meyer, G., Främling, K. and Holmström, J. (2009)** ‘Intelligent Products: A survey’, *Computers in Industry*, vol. 60, no. 3, pp. 137–148.
- Morariu, C., Morariu, O., Borangiu, T. and Sallez, Y. (2013)** ‘Formalized information representation for intelligent products in service-oriented manufacturing’, *IFAC Proceedings Volumes*, vol.46, no. 7, pp. 318-323.
- Le Mortellec, A., Clarhaut, J., Sallze, Y., Berger, T. and Trentesaux, D. (2013)** ‘Embedded holonic fault diagnosis of complex transportation systems’, *Engineering Applications of Artificial Intelligence*. Elsevier, vol. 26, no. 1, pp. 227–240.
- Naeem, M., Iianko, K., Karmokar, A., Anpalagan, A. and Jaseemuddin, M. (2013)** ‘Energy-efficient

cognitive radio sensor networks: Parametric and convex transformations’, *Sensors*, vol. 13, no. 8, pp. 11032–11050.

**Ndawi, D. B. (2015)** ‘An Improved Pegasus Protocol for Energy Efficient Wireless Sensor Network by Ant Colony Optimization’, *International Journal & Magazine of Engineering, Technology, Management and Research*, pp. 1199–1211.

**Nurdin, Y. and Muchallil, S. (2019)** ‘Holonc Multi-Agent System for Microgrid Hierarchical Control’, *Proceedings - IEEE International Conference on Cybernetics and Computational Intelligence*, IEEE, 2019, pp. 66–71.

**Occello, M., Jamont, J. P., Ben-Yelles, C. B. and Hoang, T.T.H. (2019)** ‘A multi-level generic multi-agent architecture for supervision of collective cyber-physical systems’, *International Journal of Autonomous and Adaptive Communications Systems*, vol. 12, no. 2, pp. 109–128.

**Ogundile, O. O. and Alfa, A. S. (2017)** ‘A survey on an energy-efficient and energy-balanced routing protocol for wireless sensor networks’, *Sensors*, vol. 17, no. 5, pp. 1–51.

**Paek, J., Hicks, J., Coe, S. and Govindan, R. (2014)** ‘Image-based environmental monitoring sensor application using an embedded wireless sensor network’, *Sensors*, vol. 14, no. 9, pp. 15981–16002..

**Ping, S. (2003)** ‘Delay measurement time synchronization for wireless sensor networks’, *IRB-TR-03-013, Intel Research Berkeley Lab*, pp. 12.

**Pujo, P., Broissin, N. and Ounnar, F. (2009)** ‘PROSIS: An isoarchic structure for HMS control’, *Engineering Applications of Artificial Intelligence*. Elsevier, vol. 22, no. 7, pp. 1034–1045.

**Rachuri, S., Subrahmanian, E., Bouras, A., Fenves, S. J., Fofou, S. and Sriram, D. (2008)** ‘Information sharing and exchange in the context of product lifecycle management: Role of standards’, *Computer Aided Design*, Elsevier, vol. 40, no. 7, pp. 789–800.

**Ramluckun, N. and Bassoo, V. (2018)** ‘Energy-efficient chain-cluster based intelligent routing technique for Wireless Sensor Networks’, *Applied Computing and Informatics*. vol. 16, pp. 39-57.

**Ranasinghe, D. C., Harrison, M., Främling, K. and McFarlane, D. (2010)** ‘Enabling through life product-instance management: Solutions and challenges’, *Journal of Network and Computer Applications*. Elsevier, vol. 34, no. 3, pp. 1015–1031.

**Rani, S., Malhotra, J. and Talwar, R. (2015)** ‘Energy efficient chain based cooperative routing protocol for WSN’, *Applied Soft Computing Journal*. Elsevier, vol. 35, pp. 386–397.

**Rault, T. (2017)** ‘Energy-Efficiency in Wireless Sensor Networks’, *Thèse de doctorat, Université de Technologies de Compiègne*, 2015.

**Ren, J., Zhang, Y., Zhang, K., Liu, a., Chen, J. and Shen, X. S. (2016)** ‘Lifetime and Energy Hole Evolution Analysis in Data-Gathering Wireless Sensor Networks’, *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 788–800.

**Sallez, Y., Montreuil, B. and Ballot, E. (2015)** ‘On the activeness of physical internet containers’, *Studies in Computational Intelligence*. Elsevier, vol. 594, pp. 259–269.

- Sardouk, A., Mansouri, M., Merghem-Boulaïhia, L., Gaïti, D. and Rahim-Amoud, R. (2013)** ‘Crisis management using MAS-based wireless sensor networks’, *Computer Networks*. Elsevier, vol. 57, no. 1, pp. 29–45.
- Sheltami, T., Musaddiq, M. and Shakshuki, E. (2016)** ‘Data compression techniques in Wireless Sensor Networks’, *Future Generation Computer Systems*. Elsevier, vol. 64, pp. 151–162.
- Singh, S. K., Kumar, P. and Singh, J. P. (2017)** ‘A Survey on Successors of LEACH Protocol’, *IEEE Access*. IEEE, vol. 5, pp. 4298–4328.
- Srbinovska, M., Dimcev, V. and Gavrovski, C. (2017)** ‘Energy consumption estimation of wireless sensor networks in greenhouse crop production’, *Proceedings - 17th IEEE International Conference on Smart Technologies*, 2017, pp. 870–875.
- Sun, J. Z., and Zhou, J. (2008)** ‘Power-aware data reduction for continuous query in wireless sensor networks’, *Proceedings - the IEEE International Conference on Industrial Technology*. 2008, pp. 1-6.
- Tan, H. Ö. and Körpeoğlu, I. (2005)** ‘Power efficient data gathering and aggregation in wireless sensor networks’, *ACM SIGMOD Record*, vol. 32, no. 4, pp. 66.
- Teng, H., Liu, X. and Liu, A. (2018)** ‘Adaptive Transmission Power Control for Reliable Data Forwarding in Sensor Based Networks’, *Wireless Communications and Mobile Computing*, vol. 2018. pp. 22.
- Thomas, A. (2009)** ‘RFID et nouvelles technologies de communication; enjeux économiques incontournables et problèmes d'éthique’, *Proceedings - 6ème Conférence Internationale Conception et Production Intégrées*, 2009.
- Vanjale, M., Chitode, J. S. and Gaikwad, S. P. (2020)** ‘Lifetime Estimation and Measurement for Wireless Ad Hoc Networks’, *Wireless Personal Communications*. Springer, vol. 113, no. 1, pp. 617–631.
- Ventä, O. (2007)** ‘*Intelligent products and systems: Technology theme - Final report*’, VTT Publications.
- Virmani, D., Sharma, T. and Sharma, R. (2013)** ‘Adaptive Energy Aware Data Aggregation Tree for Wireless Sensor Networks’. *arXiv*.
- Vo, D. A., Drogoul, A. and Zucker, J. D. (2012)** ‘An operational meta-model for handling multiple scales in agent-based simulations’, *Proceedings - IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future*, 2012, pp. 1-6.
- Vrba, P., Tichý, P., Mařík, V., Hall, K. H., Staron, R. J., Maturana, F. P., and Kadera, P. (2011)** ‘Rockwell automation’s holonic and multiagent control systems compendium’, *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 41, no. 1, pp. 14–30.
- Wang, Y. C. and Pan, B. C. (2020)** ‘Event-aware hierarchical routing with differential compression to extend WSN lifetime’, *Proceedings - 21st Asia-Pacific Network Operations and Management Symposium: Towards Service and Networking Intelligence for Humanity*, 2020, pp. 79–84.
- Wei, G., Ling, Y., Guo, B., Xiao, B. and Vasilakos, A. (2011)** ‘Prediction-based data aggregation in wireless sensor networks: Combining grey model and Kalman Filter’, *Computer Communications*. Elsevier, vol. 34, no. 6, pp. 793–802.

**Wong, C. Y., McFarlane, D., Ahmad Zaharudin, A. and Agarwal, V. (2002)** ‘The intelligent product driven supply chain’, *Proceedings - the IEEE International Conference on Systems, Man and Cybernetics*, 2002, vol. 4, pp. 393–398.

**Wooldridge, M. (2009)** *An Introduction to MultiAgent Systems*.

**Wooldridge, M. and Jennings, N. R. (1995)** ‘Intelligent agents: theory and practice’, *The Knowledge Engineering Review*., vol. 10, no. 2, pp. 115–52.

**World wide article information (1992)**. Available at: <http://www.wwai.org/>.

**Xiao, X., He, Q., Fu, Z., Xu, M. and Zhang, X. (2016)** ‘Applying CS and WSN methods for improving efficiency of frozen and chilled aquatic products monitoring system in cold chain logistics’, *Food Control*, vol. 60, pp. 656–666.

**Yao, Y. and Gehrke, J. (2002)** ‘The cougar approach to in-network query processing in sensor networks’, *ACM SIGMOD Record*, vol. 31, no. 3, pp. 9–18.

**Yetgin, H., Cheung, K. T. K., El-Hajjar, M. and Hanzo, L. H. (2017)** ‘A Survey of Network Lifetime Maximization Techniques in Wireless Sensor Networks’, *IEEE Communications Surveys and Tutorials*, vol. 19, no. 2, pp. 828–854.

**Younis, O. and Fahmy, S. (2004)** ‘HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks’, *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379.

**Zbib, N. (2011)** ‘Allocation et routage dynamique dans un FMS basés sur le concept de champ de potentiel’, *Thèse de doctorant. Université de Valenciennes et du Hainaut Cambrésis*, 2011.

**Zheng, G. and Hu, Z. (2010)** ‘Chain Routing Based on Coordinates-Oriented Clustering Strategy in WSNs’, *Proceedings - International Symposium on Computer Network and Multimedia Technology*, vol. 2, no. 1, pp. 1–4.

**Zhou, H., Luo, D., Gao, Y. and Zuo, D. (2011)** ‘Modeling of Node Energy Consumption for Wireless Sensor Networks’, *Wireless Sensor Network*, vol. 3, no.1, pp. 18–23.



## Résumé de la thèse en Français

Le développement rapide de l'Internet des objets conduit au paradigme du produit intelligent (Meyer et al., 2009). Équipé avec des capteurs, des modules de communication et des puces informatiques, le produit intelligent dispose de capacités de stockage et de traitement de données. Les informations relatives au produit sont accessibles pour l'utilisateur directement via RFID (Radio Frequency Identification), NFC (Near-field communication) ou à distance via d'autres technologies sans fil et Internet. L'utilisation de nœuds capteurs intelligents permet de surveiller l'environnement ou d'effectuer des tâches spéciales. En raison de l'accessibilité des données et des capacités de calcul, le paradigme du produit intelligent a été largement appliqué pour la fabrication, la logistique ou la surveillance de la santé des bâtiments (McFarlane et al., 2002) (Van Belle et al., 2011) (Lamonaca et al., 2018). Cependant, il existe certaines limites à l'utilisation d'un produit intelligent : premièrement, le flux d'informations est interrompu si l'appareil est endommagé; deuxièmement, la capacité de stockage au sein d'un produit est limitée et il est difficile de garantir que toutes les informations sur le produit pourront être stockées localement tout au long de son cycle de vie. Pour dépasser ces limites, cette thèse considère le paradigme du matériau communicant.

En 2009, le CRAN a commencé à étudier le concept de « matériau communicant » (CM). C'est un matériau qui peut traiter, stocker des données et communiquer avec l'environnement (Kubler et al., 2010). Le CM est un paradigme qui intègre la capacité de communication dans le matériel au lieu d'équiper un dispositif de communication externe. Il peut surveiller son environnement, stocker ses données monitorées et les informations relatives au produit. En outre, un CM peut soit coopérer avec d'autres pour former un CM composé, soit être décomposé en différentes parties, le CM composé ou décomposé restant toujours un matériau communicant. Le concept a été appliqué dans différents travaux, tels que (Kubler et al., 2012) (Krommenacker et al., 2013) (Jover et al., 2013) (Mekki, 2016). Ces derniers travaux ont été proposés principalement pour les applications de traçabilité des textiles, ou pour la diffusion/réplication de données dans les bétons tout au long de leur cycle de vie. Ces travaux mettent l'accent sur la capacité de communication.

Cette thèse s'intègre dans le cadre du projet McBIM (Matériau communicant avec le BIM – Building Information Modelling) (ANR Project McBIM, 2017). Basé sur le concept de matériau communicant, le projet McBIM vise à concevoir un « béton communicant » équipé

d'un réseau de capteurs sans fil (RCSF) embarqué. Ce projet est financé par l'ANR (Agence Nationale de la Recherche) et coordonné par le CRAN (Centre de Recherche en Automatique de Nancy) depuis le 1er janvier 2018. Le consortium est composé de quatre partenaires : le CRAN, deux autres unités de recherche (LAAS : Laboratoire d'Analyse et d'Architecture des Systèmes; LIB : Laboratoire d'Informatique de Bourgogne) et une entreprise (360 Smart Connect/FINAO SAS).

Le projet McBIM a deux objectifs principaux. Le premier est de concevoir le béton communicant qui permet de surveiller les paramètres physiques, de stocker, de traiter les données collectées et de transmettre périodiquement des informations aux utilisateurs. Le RCSF interne se compose de deux types de nœuds : le nœud sensitif (NS) et le nœud de communication (NC). Les NS surveillent des paramètres internes du matériau, et toutes les données physiques intéressantes (telles que la température, l'humidité ou les contraintes, etc.) sont transmises à leur NC dédié. Les NC agrègent ou traitent les données de surveillance reçues et envoient le message résultant directement ou indirectement (via d'autres NC) à une station de base.

Le deuxième objectif est de démontrer l'utilité de ce concept sur l'ensemble du cycle de vie du bâtiment. Dans différentes phases, les bétons McBIM peuvent être exploités ou utilisés par différentes entreprises avec leurs propres données BIM. En conséquence, leurs comportements de surveillance peuvent également être différents (par exemple la fréquence de surveillance). L'exigence de service à long terme, le besoin d'une grande capacité de stockage de données et d'une grande flexibilité ont conduit à différents défis.

Les différents partenaires collaborent pour atteindre ces objectifs. Le LAAS conçoit les nœuds sensitifs et de communication, le LIB étudie l'interopérabilité des données avec les applications BIM. Le 360 Smart Connect/FINAO met en œuvre le résultat de la recherche des partenaires en application réelle. Le CRAN travaille sur les couches intermédiaires entre le matériel (développé par LAAS) et l'application (développée par LIB et 360SC). Dans le cadre de ce projet, cette thèse porte sur la gestion de l'énergie pour la collecte de données périodiques réalisée par le RCSF embarqué.

Le béton communicant devant fonctionner au moins dix ou vingt ans, un élément peut coopérer avec les autres pour économiser de l'énergie. En raison de la limitation des capacités de stockage

et de calcul du béton communicant, ce travail explore le couplage de cet artefact physique avec une représentation digitale, sous la forme d'un système multi-agents, pour gérer les données et l'énergie du réseau embarqué. En effet, les nœuds communicants dans la partie physique peuvent être représentés par des agents dédiés dans la partie digitale qui peuvent non seulement stocker les informations de suivi liées au produit, mais aussi évaluer et estimer les stratégies de collecte de données pour fournir une solution optimale avec une vue globale. Cette partie digitale du béton communicant peut être vue comme un jumeau numérique.

Pour parvenir à une gestion efficace de l'énergie et prolonger la durée de vie du réseau, quatre défis sont à relever : deux pour la partie physique (questions 1 et 2, liées au RCSF embarqué), et deux associés à la partie digitale (questions 3 et 4, concernant la gestion des données et de l'énergie) comme suit :

1. Comment maximiser la durée de vie de la collecte de données (processus) au sein du RCSF intégré? Les techniques de traitement des données, la gestion du duty-cycle des nœuds et les stratégies de collecte de données entre les nœuds doivent être prises en compte pour réduire les coûts énergétiques pour la collecte de données.
2. Comment prédire l'énergie restante du réseau dans un environnement à ressources limitées? Pour prédire l'énergie restante, des modèles de consommation doivent être établis. Est-il possible de construire un modèle générique ou est-il nécessaire de spécifier des modèles énergétiques pour chaque type de structures RCSF?
3. Comment réaliser le lien entre les matériaux communicants avec leurs agents dédiés, ainsi que la gestion des données et de l'énergie des nœuds chez ces agents? La partie physique est liée à une partie digitale représentée dans cet mémoire par un système multi-agents (SMA). En conséquence, cette partie étudie la relation entre les nœuds physiques réels et la constitution de l'architecture des agents dans le SMA.
4. Comment réaliser la composition des agents? Un matériau communicant est un produit qui doit prendre en charge des caractéristiques de composition/décomposition. Cela pourrait être possible au niveau physique et au niveau digitale. Pour construire ce mécanisme, les interactions entre agents doivent être considérées.

Autour de ces défis évoqués, les principaux apports de cette étude, tant théoriques que pratiques, sont les suivants :

- Des modèles analytiques de consommation d'énergie pour la collecte périodique de données dans RCSF sont proposés. Un modèle énergétique de collecte de données est

détaillé, ainsi que le modèle de collecte de données avec ou sans agrégation (voir chapitre 3). Ces modèles prédictifs sont testés et validés par des expérimentations (voir section 5.1 du chapitre 5).

- Une architecture SMA récursive est présentée (chapitre 4) pour construire la partie digitale associée au matériau communicant. L'architecture proposée est récursive et flexible ce qui permet la composition/décomposition de matériaux communicants. La partie digitale peut fournir le niveau d'énergie restant mais aussi apporter des solutions coopératives pour répondre à une grande flexibilité du béton communicant.
- Un simulateur de béton communicant 3D est développé avec la plateforme NetLogo, où les nœuds communicants sont représentés par des agents. Cet outil numérique permet de visualiser le niveau énergétique du béton communicant et fournit l'information de durée de vie restante (en section 5.2 du chapitre 5).

Cette thèse est organisée en cinq chapitres suivis d'une conclusion générale :

Le chapitre 1 présente le contexte et les concepts du matériel communicant, ses caractéristiques et son intérêt dans le cadre du projet McBIM. Cela commence par le contexte de la gestion du cycle de vie des produits. Le produit intelligent et le paradigme holonique sont ensuite décrits. Pour surmonter les limitations du produit intelligent, le concept de matériau communicant est proposé. Afin de gérer efficacement et de prolonger le cycle de vie du béton communicant, les solutions en RCSF sont discutées, tandis qu'une architecture holonique est proposée pour la gestion des données et de l'énergie. Enfin, les enjeux de recherche de cette thèse sont détaillés.

Le chapitre 2 présente un état de l'art tant sur la partie physique que sur la partie digitale. Pour la partie physique, des solutions bibliographiques pour le traitement des données en réseau dans les RCSF sont présentées : fonctions d'agrégation de données, protocoles de routage efficaces, protocoles de synchronisation et modèles de consommation d'énergie. Pour la partie digitale, l'architecture holonique traditionnelle dédiée au produit intelligent est présentée, ainsi que les approches multi-agents pour la gestion de l'information. De plus, pour répondre aux exigences de composition/décomposition de l'application CM, des solutions récursives sont examinées et discutées.

Le chapitre 3, lié à la littérature sur la partie physique, présente un modèle de consommation d'énergie basé puissance pour le nœud communicant. En se basant sur cette représentation, des modèles énergétiques analytiques sont proposés pour estimer la consommation d'énergie pour

la collecte de données avec/sans agrégation dans un RCSF. En outre, les modèles d'énergie analytiques pour la collecte d'agrégation de données avec ou sans perte sont également abordés pour tout type de structure de topologie RCSF.

Le chapitre 4, relatif à la littérature sur la partie digitale, détaille l'architecture réursive du système multi-agents (SMA) proposée et le modèle d'agent du matériau communicant. En outre, les mécanismes de composition/décomposition sont discutés, la gestion et le partage d'informations entre les agents sont détaillés en particulier pour la gestion de l'énergie ainsi que l'utilisation de cette approche pour estimer la durée de vie d'éléments et la composition/décomposition.

Le chapitre 5 présente les résultats applicatifs et démonstratifs tant pour la partie physique (RCSF) que digitale (SMA). Les modèles énergétiques analytiques pour la collecte de données avec/sans agrégation sont validés avec une plateforme Arduino-XBee (dans la section 5.1). Sur la partie digitale, un modèle 3D basé sur des agents est développé sur la plate-forme Netlogo. Ce modèle peut être utilisé pour visualiser le niveau d'énergie du nœud de communication, analyser et fournir une solution optimale pour prolonger la durée de vie de la partie physique.

Enfin, la thèse se termine par une conclusion générale dans laquelle les contributions de nos travaux sont résumées, et quelques perspectives sont discutées.