SPIM
Thèse de Doctorat

école doctorale sciences pour l'ingénieur et microtechniques
UNIVERSITÉ DE FRANCHE-COMTÉ

Energy Efficient Data Handling and Coverage for Wireless Sensor Networks

■ Hassan Saïd MOUSTAFA HARB

SPIM
Thèse de Doctorat

école doctorale sciences pour l'ingénieur et microtechniques
UNIVERSITÉ DE FRANCHE-COMTÉ

Energy Efficient Data Handling and Coverage for Wireless Sensor Networks

By

Hassan Saïd MOUSTAFA HARB

A Dissertation Submitted to the

University of Franche-Comté

in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in Computer Science

in July 12, 2016

Dissertation Committee:

Pr. Richard CHBEIR	University of Pau	Reviewer
Pr. Hamamache KHEDDOUCI	University of Claude Bernard Lyon 1	Reviewer
Pr. YE-QIONG SONG	University of Lorraine	Examinator
Dr. Bechara AL BOUNA	Antonine University	Examinator
Pr. Raphaël COUTURIER	University of Franche-Comté	Supervisor
Dr. Abdallah MAKHOUL	University of Franche-Comté	Co-supervisor
Pr. Oussama BAZZI	Lebanese University	Supervisor
Dr. Ali JABER	Lebanese University	Co-supervisor

Abstract

Energy Efficient Data Handling and Coverage for Wireless Sensor Networks

Hassan Saïd Moustafa Harb
University of Franche-Comté, 2016

Supervisors: Raphaël Couturier, Abdallah Makhoul, Oussama Bazzi and Ali Jaber

Wireless sensor networks (WSNs) have become a highly active research today. Their applicability can be seen diverse domains: environmental and habitat monitoring, military surveillance, industrial process controlling, aquatic environment observing, natural disaster prevention, etc. In general, a WSN consists of a large number of small sensing self-powered autonomous nodes (sensors) that collect information about the monitored phenomenon. The sensors send the collected data, via wireless communication, to a sink node either directly or through intermediate nodes. Sensor nodes are very small devices with limited resources such as memory, battery and computation power. Therefore, the main factor affecting nodes' lifetime is their limited battery energy, which usually highly depends on data sensing and transmission power consumption. In addition, in many applications, the sensor nodes are scattered in dangerous or inaccessible areas and left unattended. Thus, replenishing or replacing their battery is extremely difficult if not impossible. Hence, reducing the energy consumption in sensor networks is the major challenge for research in order to increase network lifetime.

In this thesis, we are interested in the periodic data collection model in WSN, which we call periodic sensor networks (PSNs), based on the clustering architecture of the network. In such network, each sensor node monitors the given area and sends its collected data periodically (at each period) to the sink via its proper cluster-head (CH). Consequently, PSN faces two major challenges. First, it offers a great amount of collected data and thus enables complex data analysis for decision makers. Second, the energy of sensors will be depleted quickly due to the huge volume of data collection and transmission. Therefore, researchers' strategies are often targeted to minimize the amount of data retrieved/communicated by the network without considerable loss in fidelity. The goal of this reduction is first to increase the network lifetime, by optimizing energy consumption of the limited battery for each sensor node, and then to help in analyzing data and making decision. In this thesis, we propose energy-efficient data management techniques dedicated to periodic sensor networks (PSNs) based on clustering architecture. More specifically, we focus on data collection, data aggregation and data correlation in PSNs with the main goal of extending the network lifetime.

First, we propose an adaptive data collection mechanism in order to minimize the amount of data collected by each sensor during the data collection phase in PSN. Our objective is to allow each sensor node to adapt its sampling rate to the changing of the monitored condition. We study the sensed data between periods based on the dependence of conditional variance on measurements varies over time with three different tests (Fisher, Tukey and Bartlett). Then, we use an existing multiple levels activity model that uses behavior functions modeled by modified Bezier curves to define application classes and allow each node to compute its sampling rate while taking into account its residual energy level.

The second objective of this thesis is to eliminate redundant data generated in each cluster at both sensors and CH levels. At the first level, each sensor node searches the similarities between readings collected at each period in order to eliminate redundancy from raw data. Then, it searches duplicated data sets captured among successive periods, using the sets similarity functions, in order to reduce data sets transmission to the its CH. At the second level, we propose a data aggregation technique based on the distance functions in order to allow CH to find, then eliminate, redundant data sets generated by neighboring nodes, before sending final sets to the sink.

Third, we exploit the spatio-temporal correlation between sensor nodes in order to optimize the coverage of the zone of interest. Based on this correlation, we propose two sleep/active strategies for scheduling sensors in each cluster. The first one searches the minimum number of active sensors, e.g. which they will collect data, based on the set covering problem while the second one takes advantages from the correlation degree and the sensors residual energies for scheduling nodes in the cluster.

To evaluate the performance of the proposed techniques, simulations on real data collected from 54 sensors deployed in the Intel Berkeley Research Lab have been conducted. We have analyzed their performances according to energy consumption, data latency and accuracy, and area coverage, and we show how our techniques can significantly improve the performance of sensor networks.

KEYWORDS: Periodic Sensor Networks, Clustering Architecture, Adaptive Sensor Sampling Rate, Similarity and Distance Functions, Spatio-Temporal Correlation, Coverage and Scheduling Strategies.

RÉSUMÉ

Gestion Efficace de Données et Couverture dans les Réseaux de Capteurs Sans Fil

Hassan Saïd Moustafa Harb
Université de Franche-Comté, 2016

Encadrants: Raphaël Couturier, Oussama Bazzi, Abdallah Makhoul et Ali Jaber

Les réseaux de capteurs sans fils (RCSF) est une recherche très active aujourd'hui. Ils sont applicables dans divers domaines: la surveillance de l'environnement et de l'habitat, la surveillance militaire, le contrôle des processus industriels, l'observation des milieux aquatiques, la prévention des catastrophes naturelles, etc. En général, un RCSF se compose d'un grand nombre de petits nœuds autonomes et auto-alimentés (capteurs) qui collectent des informations sur une zone surveillée. Les capteurs envoient les données collectées, via la communication sans fil, vers le puits, soit directement ou à partir des nœuds intermédiaires. Les capteurs sont des appareils très petits avec des ressources limitées telles que la mémoire, la batterie et la puissance de calcul. Par conséquent, le principal facteur affectant la durée de vie des capteurs est leur énergie. Elle est limitée dans la batterie et elle dépend fortement de la collecte et la transmission de données. En outre, dans de nombreuses applications, les capteurs sont dispersés dans des zones dangereuses ou inaccessibles. Ainsi, renouveler ou remplacer leur batterie est extrêmement difficile, sinon impossible. Par conséquent, la réduction de la consommation d'énergie dans les réseaux de capteurs représente le principal défi pour les chercheurs en vue d'augmenter la durée de vie du réseau.

Dans cette thèse, nous nous sommes intéressés au modèle de collecte de données périodique dans les réseaux de capteurs, que nous appelons les réseaux de capteurs périodiques (RCPs), basé sur l'architecture clustering du réseau. Dans un tel réseau, chaque capteur surveille la zone cible et envoie ses données collectées périodiquement (à chaque période) au puits via son cluster-head (CH) approprié. En général, le RCP est confronté à deux défis majeurs. Premièrement, il offre une grande quantité de données collectées qui complique l'analyse de données des décideurs. Deuxièmement, l'énergie des capteurs s'épuise rapidement en raison de l'énorme volume de données collectées et envoyées. Par conséquent, les stratégies des chercheurs sont souvent ciblés pour réduire au minimum la quantité de données collectées/communiquées dans le réseau sans perte considérable de leur fidélité. Le but de cette réduction est d'abord d'augmenter la durée de vie du réseau, en optimisant la consommation d'énergie de la batterie limitée pour chaque capteur, puis pour aider à l'analyse de données et la prise de décision. Dans cette thèse, nous proposons des techniques de gestion de données pour économiser la consommation de l'énergie dans les réseaux de capteurs périodiques (RCPs) basés sur l'architecture de clustering. Plus précisément, nous nous concentrons sur la collecte de données, l'agrégation de données et les corrélations de données dans RCPs dont l'objectif principal de prolonger la durée de vie du réseau.

Le premier objectif de cette thèse est de minimiser la quantité de données collectées dans chaque capteur pendant la phase de la collecte de données dans le RCS. Nous permettons à chaque capteur d'adapter son taux d'échantillonnage en se basant sur le changement de la condition surveillée. Tout d'abord, nous étudions les données collectées dans chaque période en fonction de la dépendance de la variance de données qui varient dans le temps avec trois tests différents (Fisher, Tukey et Bartlett). Ensuite, nous proposons un modèle d'activité à multiple niveaux, qui utilise des fonctions de comportement modélisées par des courbes de Bézier, pour définir les classes de criticité des applications et permettre à chaque capteur de calculer son taux d'échantillonnage tout en tenant compte de son niveau d'énergie résiduelle.

Le deuxième objectif de cette thèse est d'éliminer les données redondantes générées dans chaque cluster au niveau des capteurs et CH. Au premier niveau, chaque capteur cherche la similarité entre les données collectées à chaque période afin d'éliminer la redondance entre les données brutes. Ensuite, il cherche les ensembles de données dupliquées capturées dans les périodes successives, en utilisant des fonctions de similarité, afin de réduire le nombre des ensembles de données envoyées au CH. Au deuxième niveau, nous proposons une technique d'agrégation de données basée sur les fonctions de distance qui permet au CH de trouver, puis éliminer, les ensembles de données redondantes générées par les nœuds voisins, avant d'envoyer les ensembles finaux au puits.

Le troisième objectif de cette thèse est de chercher la corrélation spatio-temporelle entre les nœuds capteurs pour exploiter la redondance existant dans le réseau. Sur la base de cette corrélation, nous proposons deux stratégies d'ordonnancement actif/inactif pour ordonnancer les capteurs dans chaque cluster. La première stratégie cherche le nombre minimal de capteurs actifs, i.e. qui doivent collecter les données, en se basant sur le problème de couverture des ensembles tandis que la deuxième stratégie prend avantages du degré de corrélation et les énergies résiduelles de capteurs pour ordonnancer les nœuds capteurs dans chaque cluster.

Pour évaluer la performance des techniques proposées, des simulations sur des données réelles collectées à partir de 54 capteurs déployés dans le laboratoire Intel Berkeley Research Lab ont été menées. Nous avons analysé leurs performances selon la consommation d'énergie, la latence et l'exactitude des données et la couverture de la zone surveillée, et nous montrons comment nos techniques peuvent améliorer considérablement les performances des réseaux de capteurs.

MOTS-CLÉS: Réseaux de Capteurs Périodiques, Architecture Clustering, Adaptation de Taux d'échantillonnage de Capteurs, Fonctions de Similarité et de Distance, Corrélation spatio-temporelle, Couverture et Stratégies d'Ordonnancement.

Contents

List of Figures

List of Tables

LIST OF ALGORITHMS

ABBREVIATIONS

WSN	Wireless Sensor Network
PSN	Periodic Sensor Network
ANOVA	ANalysis Of VAriance
CH	Cluster-Head
$\mathbf{S_i}$	Sensor node number i
$\mathbf{TV}$	Total Variation
$\mathbf{VWP}$	Variation Within Period
$\mathbf{VBP}$	Variation Between Period
BV	**B**eha**V**ior Function
$\mathbf{r^0}$	Application Criticality Level
$\mathbf{p_i}$	Period number i
$\mathbf{E_{r_i}}$	Residual Energy of the Sensor i
$\mathbf{ST}$	Sensor Sampling Rate
$\mathbf{S_{max}}$	Maximum Sensor Sampling Rate
PFF	Prefix-Frequency Filtering Technique
$\mathbf{N}$	Total Number of Sensor Nodes
$\mathbf{n_i}$	Number of Sensor Nodes in the Cluster Number i
$\mathbf{R_i}$	Vector of Readings Captured during a Period by the Sensor Node i
$\mathbf{R'_i}$	Set of Readings After Applying $Similar$ Function over the Vector R_i
$\mathbf{r}$	A Reading Captured by a Sensor
$\mathbf{wgt(r)}$	Weight for a Reading
$\mathcal{T}$	Number of Measures during a Period
δ	Threshold for the $Similar$ or $LocTmp$ Functions
$\mathbf{t}$	Jaccard Similarity Threshold
$\mathbf{t_d}$	Distance Threshold
$\mathbf{E_d}$	Euclidean Distance between Two Data Sets

$\mathbf{C_d}$	Cosine Distance between Two Data Sets
$\mathbf{E_g}$	Geographical Euclidean Distance between Two Sensors
$\mathbf{S_r}$	Sensing Range for a Sensor
$\mathbf{C_{sp}}$	Spatial Correlation Threshold
$\mathbf{C_{tp}}$	Temporal Correlation Threshold
$\mathbf{C_{sptp}}$	Spatio-Temporal Correlation Threshold
PPMC	Pearson Product-Moment Coefficient Technique
$\mathbf{t_{PPMC}}$	Similarity Threshold for PPMC
SC	Set Cover Strategy
CDRE	Correlation Degree and Residual Energy Strategy

DEDICATION

For the sake of **ALLAH** (Subhanahu Wa Ta'ala) and due to her unique biography, I am dedicating this thesis to the Lady Fatima Al-Zahra (sa), the Greatest of Women of the Heavens and the Earth.

Fatima Al-Zahra (sa) belongs to the noblest family to have existed throughout the history of mankind. Due to the unique care and love she showed her distinguish father, the Prophet Muhammad (Peace and Blessings upon him and His Family), Lady Fatima (sa) was referred as "*The Mother of her Father*" giving the humanity how should the relatioship between childs and their parents.

Lady Fatima (sa) had married to the Leader of all Believers, Ameerul Momineen, Ali ibn Abi Talib (as) in a very simple mariage traditions proved that the couple life should be based on love and respect and not benefit or money, while giving the humanity a beautiful paradigm on how should the relationship between the maried.

Lady Fatima (sa) is the mother of the Masters of the youths of heaven (*Syed-e-Shabab-e-Ahlul Jannah*), Imam Hassan (as) and Husayn (as), and grandmother of the other nine Imams of Islam. These Imams, with unparalleled virtues and merits, have teaching the humanity the lecons for all things in the life.

Even after the Holy Prophet's death, Lady Fatimah (sa) displayed her bravery, valor and courage by challenging those who had usurped her rights and the Caliphate of her husband. Although she also lost the last of her sons, Mohsin (who died as an unborn foetus), Lady Fatimah (sa) had stay patient and side her husband in the most difficult moments.

For all the mentioned reasons, I dedicated this thesis to the Lady Fatima Al Zahra (sa) as the best paradigm to be followed in our life.

ACKNOWLEDGEMENTS

First and foremost I would like to thank **ALLAH** (Subhanahu Wa Ta'ala), the biggest source of happiness and knowlegde, whose many blessings have made me who I am today. I could never have done this without the faith I have in you. Then, I would like to thank all the people who contributed in some way to the work described in this thesis.

I would like to express my sincere appreciation and gratitude to my supervisors: Prof. Raphaël Couturier and Dr. Abdallah Makhoul, at the University of Franche-Comté (UFC), and Prof. Oussama Bazzi and Dr. Ali Jaber, at the Lebanese University (UL), for their guidance during my research. Their support, efforts and inspiring suggestions were essential to the birth of this document and to my formation as a future researcher. Particulary, I am very grateful to **Dr. Abdallah Makhoul** who has been a constant source of encouragement and enthusiasm, not only during this thesis but also during my Master intership. I appreciate all his contributions of times, ideas, and funding to make my Ph.D productive and stimulating, only my forever prayer will be sufficient to thank him.

For this thesis, I am grateful to my reading committee members Prof. Richard Chbeir and Prof. Hamamache Kheddoucci for their time, interest, and helpful comments. I would also like to thank the other two members of my oral defense committee, Prof. Ye-Qiong Song and Dr. Bechara al Bouna, for their time and insightful questions.

I would like to gratefully acknowledge the Islamic Center Association for Guidance and Higher Education, Liban for financial support. As well as, my Ph.D was work possible thanks to the support of AL Rayan restaurant, France represented by M. Hassan Abou Hamdan during my stay in France.

I am very gratefully to all people I have met along the way and have contributed to the developement of my research. My appreciation and thanks go to all members of the Femto-St laboratory at UFC and all members of the comupter science department at UL. I will forever remember in my heart the stories and the moments you shared with me during these years together. I would also like to express my thanks to Mdm. Ingrid Couturier for all the received assistance during my study.

Lastly, my deepest gratitude goes to my family for their unflagging love and unconditional support throughout my life and my studies. For my parents, Saïd and Mariam, who raised me with a love of science and supported me in all my pursuits. You made me live the most unique, magic and carefree childhood that has made me who I am now. For my brothers Ali and Mohammad, and my sisters Jamila and Batoul whose faithful support during this Ph.D. The support and the encouraging of all members of my big family are so appreciated. **THANK YOU**.

Introduction

1. General Introduction

Recently, Wireless Sensor Networks (WSNs) have been defined, according to MIT Technology Review, one of 10 emerging technologies that will change the world [30]. Therefore, they have attracted a great attention of researchers and they have become one of the most interesting areas of research in the few years. Typically, a WSN is composed of a huge number of distributed sensor nodes with the aim of monitoring a physical condition from remote locations. As sensor node is defined as small, low-cost and limited resources device that communicates wirelessly and has the capabilities of sensing, processing, storing and sending collected data. Currently, a huge number of WSNs have been deployed in a variety of fields for many purposes such as environmental monitoring, productivity improving, enhanced safety and security, precision agriculture and food improving, healthcare surveillance, aquatic environments monitoring, disasters prevention, etc. Therefore, various kinds of sensors have been proposed to collect different types of data such as chemical, optical, thermal, biological, acoustics, aerial, etc.

Depending on application requirements, data collection in WSNs can either be triggered by external sources, such as queries to get a snapshot view of the network, or events when they appear, or periodic monitoring without any external triggering [42, 63]. In this thesis, we focus on the last model of data collection, i.e. periodic reporting data collection, which it is called periodic sensor networks (PSNs). Sensors in such networks collect data from target area on a periodic basis and then forward them toward a specific node “the sink” at the end of each period. Such types of networks are very suitable for many applications that need continuous real-time data collection for analysis and decision purposes. On the other hand, a lot of research proposed recently have considered clustering as the most useful architecture for many applications in WSNs; in most scenarios of WSNs, a massive deployment of sensor nodes is required, thus, makes the communication management in the network is very difficult. Indeed, clustering can reduce the overall communication cost, ensure the network scalability, ensure energy efficient in finding routes and it is easy to manage. Hence, this thesis is mainly dedicated to the periodic sensor networks based on the clustering architecture.

Currently, researchers attention is focused to the most critical constraint in PSNs: The energy consumption. Mostly, sensor nodes have a non-renewable power supply and, once deployed, must work unattended. Therefore, limited energy available in the sensors must be used effectively in order to increase the network lifetime as long time as possible. It is shown that data transmission in WSNs is a very expensive operation in terms of energy consumption and it consumes the most energy in the sensors [119, 11]. In addition, PSNs generate a large amounts of data communication, due to their periodic manner in data collection, which drains quickly the energy in the network. Therefore, to address the energy constraint, research efforts have been done today to design efficient data management techniques for PSNs. The main objective of such techniques is to reduce the amount of data collected/transmitted in the network thus, minimizing the energy consumption and improving the network lifetime. Furthermore, data reduction in data management techniques can help the decision makers in analyzing data in order to better understand the monitored phenomenon, while guaranteeing the integrity of data.

In this thesis, we propose energy-efficient data management techniques dedicated to periodic sensor networks (PSNs) based on clustering architecture. More specifically, we focus on data collection, data aggregation and data correlation in PSNs with the main goal of extending the network lifetime. We are interested in studying data generated in the network in order to reduce the volume of data sent to the sink. We propose several techniques that manage data collected/transmitted in each cluster, where appropriate algorithms have been applied at sensor node and cluster-head (CH) levels. Our proposed techniques are validated via simulations on real sensor data and comparison with other existing data management techniques. The results show that the effectiveness of our techniques in terms of improving the performance of the network and extending its lifetime, while taking into account the requirements of the monitored application.

2. Main Contributions of this Thesis

The main contributions in this thesis concentrate on designing energy-efficient data collection, data aggregation and data correlation techniques for cluster-based PSNs.

A) Data Collection: Sensors are typically deployed to gather data about the monitored environment and transmit them at a fixed periodicity to the sink node. Indeed, data collected are highly dependent on how fast the physical condition or process varies and what intrinsic characteristic need to be captured. Consequently, redundant data may be collected and forwarded from a sensor node to the sink during consecutive periods. Hence, data management in PSNs should include efficient data collection techniques for an energy efficient use of the sensors. One of the fundamental mechanism for energy optimization and data reduction in periodic data collection is the adaptive sampling approach. The objective of such approach is to allow each sensor node to adapt dynamically its sampling rate according to the dynamics of the monitored environment. This reduces the amount of redundant data collected and minimizes the activity of the sensor's radio (hence saving energy) while maintaining sufficiently high quality and resolution of the collected data to enable meaningful analysis.

In this thesis, we propose an adaptive sampling approach for energy-efficient data collection in PSNs. The proposed technique allows each sensor node to adapt its sampling rate based on the dependence of conditional variance of readings that varies over time. We study three different statistical tests (Fisher, Tukey and Bartlett) based on the one-way ANOVA model while taking into consideration the residual energy of each node. Otherwise, monitored applications have not the same critical level thus, we cannot adapt the sampling rate of the sensor node in the same manner for all applications; in other words, application with high risk level requires more collected readings from the sensor than that with low risk level in order to maintain a high quality of the collected data. Therefore, in order to define the application criticality classes, we use an existing multiple level activity model [76] that uses behavior functions modeled by modified Bezier curves to allow for sampling adaptive rate.

B) Data Aggregation: Due to random and dense deployment of the network, sensor nodes provide a high redundancy in sensed data. Furthermore, the power consumption is at the highest level when sending and receiving messages thus, the volume of data transmission must be minimized. Therefore, a huge effort in research works have made on the data aggregation in sensor networks in the last decade. The objective of such technique is to eliminate redundancy and minimize the number of transmissions to the sink, thus saving energy and improving network lifetime. Indeed, data aggregation takes more attention in cluster-based PSNs; since data collected in each period are usually redundant, sensor node must eliminates the redundancy among raw data before sending them to its CH; on the other hand, CH should remove redundant data generated by neighboring nodes before sending them to the sink.

This thesis proposes a complete data aggregation framework for cluster-based PSNs. The proposed techniques aim at eliminating redundant data generated in each cluster by proposing several aggregation algorithms for sensors and CH nodes. At the sensor level, an aggregation process allows sensor node to eliminate similar readings collected in each period. Then, it allows the sensor to remove duplicated data sets captured among successive periods in order to reduce data sets transmission to the CH. At the CH level, we propose a filtering aggregation process to allow CH to find, then eliminate, redundant data sets generated by neighboring nodes, before sending final sets to the sink.

C) Data Correlation: Exploring inter-nodes correlation is a well-known strategy in sensor networks which helps to increase the battery life of sensor nodes. In PSNs, the densely deployment and the periodic collection of data make the correlation between sensor nodes be typically spatio-temporal. On the one hand, due to the geographical location of sensors, the generated sensory data by neighboring nodes are often spatially correlated. On the other hand, periodic data captured by sensor nodes are mostly temporally correlated. This correlation is due to the slow variation of the monitored phenomenon. As a result, temporal correlation can be detected at the sensor node level among its consecutive readings or at the CH level among readings collected by neighboring sensor nodes at the same period. Hence, exploring spatio-temporal between sensors have become an emerging topic in periodic sensor networks today where researchers have motivated to explore such correlation when designing data gathering mechanisms. Moreover, turning off redundant sensor nodes has been proven as efficient way to reduce data transmission and to improve power efficiency, without a large degradation of observation fidelity. Indeed, redundant nodes with high spatio-temporal correlation in their collected data can go to a sleep mode and in a periodic manner.

In this thesis, we study and propose an energy-aware spatio-temporal scheduling techniques for real-time data collection in clustering-based architecture periodic networks. First, we search the spatio-temporal correlation between neighboring nodes, based on the Euclidean distance, in order to exploit the redundancy existing in data collection. Then, we propose two scheduling algorithms to select a set of representative nodes in each cluster to collect and transmit data to the sink, and to set the remaining nodes in the cluster to the sleep mode. In the first scheduling algorithm, the representative nodes are chosen based on the set covering problem, while, in the second algorithm, they are chosen based on the correlation degree and the residual energy of the sensors.

4. Thesis Structure

The thesis is structured as follows:

Chapter 1: Periodic Wireless Sensor Networks: An Overview: This chapter provides an introduction to the wide field of periodic sensor networks. We present the various concepts related to its objectives, its features and the different fields of application. We also present the clustering scheme as an efficient architecture for periodic sensor networks. Furthermore, we describe the main challenges that face such networks, such energy consumption, data management, coverage, etc. Because energy is the primary challenge, we highlight, in this chapter, the importance of studying data collection, data aggregation and data correlation in order to reduce the power consumption in periodic sensor networks.

Chapter 2: Adaptive Real-Time Data Collection Model: This chapter focuses on the problem of big data collected in periodic sensor networks. We propose an efficient adaptive model of data collection for PSN in order to reduce the huge amount of collected data thus, increasing the network lifetime. The proposed approach allow each sensor node to adapt its sampling rate to the physical changing dynamics. We use one-way ANOVA model with three statistical tests (Fisher, Tukey and Bartlett), while taking into account the residual energy of the sensor. Then, since each application has its own level of criticality, we use an existing multiple level activity model that uses behavior functions modeled by modified Bezier curves to define application classes and allow each sensor for adapting its sampling rate.

Chapter 3: Energy-Efficient Data Aggregation and Transfer Protocol: This chapter is dedicated to reduce the energy cost of data transmission in sensor nodes. We suggest a two phase data aggregation technique based on clustering approach for energy efficiency in PSN. In the first phase, called aggregation phase, each sensor searches similarities between readings collected at each period. In the second phase, called transmission phase, the sensor reduces the number of data sets sent to its CH by searching similarity between captured readings among successive periods using sets similarity functions. .

Chapter 4: In-network Data Aggregation Technique: In this chapter, we propose a complete data framework for cluster-based PSN aiming to eliminate redundancy at both sensor nodes and CH levels. Further to a local aggregation at each sensor node, our technique allows CH to find duplicated data sets generated by neighboring sensor nodes at each period. Aggregation data phase proposed at CH level is based on distance functions such as Euclidean and Cosine. Once redundant data sets are found, CH uses a selection algorithm to select the data sets to be sent to the sink among the received data sets.

Chapter 5: Spatio-Temporal Data Correlation with Scheduling Strategies: This chapter is dedicated to explore data correlation between neighboring sensor nodes.

We propose an efficient mechanism based on the Euclidean distance for searching the spatial-temporal correlation between sensor nodes in periodic applications. Based on this correlation, we propose two sleep/active strategies for scheduling sensors in the network. The first one searches the minimum number of active sensors based on the set covering problem while the second one takes advantages from the correlation degree and the residual energy of the sensors when scheduling nodes in the cluster.

Chapter 6: Conclusion and Perspectives: This chapter concludes our work and highlights some aspects of suggested future research work.

1

Periodic Wireless Sensor Networks: An Overview

Wireless sensor networks (WSNs) have been considered as one of the most important technologies used in 21st Century. Basically, a WSN consists of a large number of sensor nodes which are densely deployed over the monitored area in order to collect data about such area. In this thesis, we focus on a specific type of WSNs which called periodic sensor networks (PSNs). In PSN, sensor nodes collect, then send, data periodically to the end user in order to perform real time data collection for the monitored area. The first chapter in this thesis gives an overview about PSNs. First, we review a number of PSN applications via some existing examples. Then, we describe challenges faced to PSNs while highlighting the data management challenge as a real problem for such networks.

1.1/ Introduction

Currently, the world faces unprecedented challenges in environmental monitoring; the sources of environmental pollution (increasing population, urbanization, transportation, etc.) and the natural disasters (floods, earthquakes, etc.) are increasing day to day. These challenges lead to large-scale impacts on the environment, such as global warming, and might affect a large number of people. Therefore, collecting and analyzing environmental data is becoming essential for decision makers in order to avoid any potential risks in the future. Consequently, wireless sensor networks (WSNs) have attracted, especially after the nuclear disaster caused by the Great East Japan Earthquake on March 2011, a significant amount of interest from many researchers as a means of realizing phenomena monitoring in a large scale area [146]. One of the advantages of these networks is their ability to operate unattended in harsh environments in which contemporary human-in-the-loop monitoring schemes are risky, inefficient and sometimes infeasible [3]. In such networks, sensors are expected to be remotely deployed, e.g. via helicopter or clustered bombs, in a wide geographical area to monitor the changes in the environment and send back the collected data to the end user.

Nowadays, Wireless Sensor Networks (WSNs) are almost everywhere and they have become one of the innovative technologies that are widely used. They are exploited for thousands of applications such as environment, industrial, agriculture, water and ocean monitoring, health-care, etc. Constantly, WSN is built of “sensor nodes” from a few to several hundreds or even thousands which are responsible for monitoring a sensor area and transmit data back to a collection point called ‘sink’. In this network, each sensor node is capable of performing sensory information, processing and communication with each others in the network without wires.

Depending on application requirements, data collection in WSNs can be categorized into three different models: query-driven model, event-driven model, or time-driven model. In query-driven model, sensor nodes send back their data in response to a received query generated by the sink in order to get a snapshot view of the network. In event-driven model, sensor nodes send their data to the sink only when an event occurs such as human intruder detection [71], pipeline vandalisation and oil spillage [53]. Finally, the time-driven delivery model is suitable for applications that require continuous periodic monitoring such as environmental and activity humans monitoring [106, 98]. In this model, sensor nodes collect data of interest and forward them to the sink at constant periodic time intervals. In this thesis, we focus on the last model of data collection, periodic reporting data collection, which we will call Periodic Sensor Networks (PSNs).

The remainder of this chapter is organized as follows. Section 1.2 introduces a periodic wireless sensor network and its main objectives. Section 1.3 review a number of PSNs applications via some existing examples. In Section 1.4, we describe the clustering architecture used for PSNs. The main challenges and the data management in PSNs are presented in Sections 1.5 and 1.6 respectively. Finally, we conclude the chapter in Section 1.7.

1.2/ Periodic Sensor Network (PSN): A Definition

By definition [120, 95], PSN is a wireless sensor network where sensors periodically collect data about the zone of interest before sending them toward the sink. Contrarily to other types of networks, PSNs have a huge capability of ensuring a continuous real time data collection of the interest zone. Furthermore, periodic sampling data model is one of the most prominent and comprehensive ways of data collection to extract raw sensor readings [46, 95]. Environmental monitoring [106] and phenomena surveillance [157] are main examples of PSNs applications where the area of interest is monitored constantly. In such applications, the common task of a sensor node is monitoring some phenomena, collecting periodically local readings of interest and relaying data toward the sink at each period. Figure 1.1 shows an example of PSN where each sensor node takes one data reading each ten minutes then send its set of collected data which contains six readings to a cluster head (CH) at the end of each hour.

1.3/ Applications

Thanks to its capabilities of continuous monitoring of large areas, periodic sensor networks (PSNs) have attracted significant attention in many applications, such as environment, industrial, underwater, medical, etc. in such applications, the main objective of PSN is to deliver periodically real time data to the end user about the region of interest. Next, we give an overview about different PSN applications while detailing some real projects.

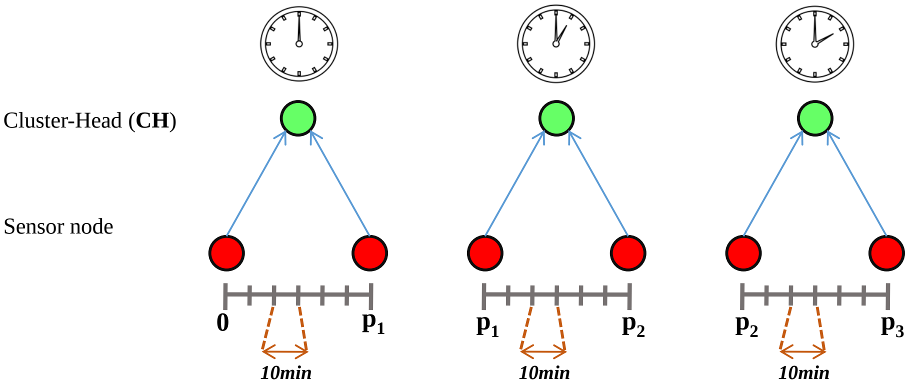


Figure 1.1: Illustrative example of periodic sensor network (PSN).

1.3.1/ Environment Monitoring

Environmental-based applications are the main domain where PSNs are broadly used. In such applications, the main objective is to monitor, observe and study the different kinds of natural environments and phenomenon, in order to enable a better understanding of such environments. Environmental monitoring based on PSNs includes weather characteristics monitoring, natural disasters monitoring, animal behavior surveillance, traffic control, smart home automation, etc.

The GAEMN project, Georgia Automated Environmental Monitoring Network, is a PSN on the purpose of monitoring the weather characteristics. In GAEMN, sensors collect data every 1 second while data are summarized and sent at every period of 15 minutes. When data are received by the sink, it process them then sends them via the internet [57]. In a latter time, data are analyzed within a cyber-infrastructure system [14]. Volcano Tungurahua [157] is another project that uses a periodic sensor networks to monitor the activity of the volcano. Such networks are constructed from small microphones in order to sense the signals generated by the volcano during its eruptions. Infrasound data gathered by the network over 54 hours are forwarded to a base station at the volcano observatory placed near 9 km of the volcano. Aiming to understand the relationship between glacier dynamics and climate change, GlacsWeb project based on the sensor networks has been developed [92]. By embedding sensors inside the glacier, GlacsWeb has allowed to collect data about different glacial characteristics such as water pressure, case stress, temperature, tilt angle and resistivity. After one year of continuous monitoring with an average of $36,078$ readings for each sensor, the decision makers were able to understand how varies the climate based on the glacier dynamics.

1.3.2/ Water and Ocean Monitoring

In the last decade, researchers have been motivated by two reasons to explore the aquatic environment; first, the ocean covers more than 70% of the Earth's surface while only less than 10% of its volume has been investigated; second, it is very important for many humanity purposes, such as nourishment production, transportation, defense and adventurous, etc. Therefore, researchers use sensor networks as one of the most important technologies for such type of environments. PSNs applications of aquatic environment include monitoring of water characteristics, seismic and tsunami, spillage oil,

Table 1.1: Environmental monitoring projects based on PSNs.

Project	Zone of interest	Description
Sonoma Dust [142]	Sonoma County, California, USA	monitoring of redwood trees habitat conditions
Lofar agro [116]	Netherlands	monitoring the microclimate in potato crops
Foxhouse [51]	Fur Farming Research Station, Kannus	monitoring the habitat of foxes in a fox house
Sensorscope [23]	Canton Valais, Switzerland	monitoring of air temperature and humidity, wind direction and speed, precipitation and solar radiation
GDI [136, 89]	South of Mount Desert Island, Main, USA	habitat monitoring in Great Duck Island
BSpringbrook [156]	South-East Queensland, Springbrook, Australia	rainforest ecosystems monitoring

pollution control and climate recording, assisted navigation and study of marine life.

The ARGO project [115] is an example of underwater PSNs. Argo deploys more than 3000 sensors distributed over the global oceans to periodically collect salinity, temperature and velocity readings from the upper 2000 meters of depth. Every ten days, data collected by the sensors are transmitted to a satellite while the nodes are always on the surface. In [77], the authors deploy a small PSN constituted of 9 sensors for water monitoring in the Burdekin delta, Australia. Sensors scattered over a region of 2-3 km^2 involves periodic data such as pressure, water flow rate, and salinity then, send them to the sink node which, in turn, forwarded them to a remote site for archiving and processing. The objective of this project is to monitor the coastal region that, over extraction of water, leads to saltwater intrusion into the aquifer. Another PSN was deployed to measure vertical temperature profile at multiple points on a large water storage that provides most of the drinking water for the city of Brisbane, Australia [148]. The data, from a string of temperature transducers at depths from 1 to 6 meters at 1 minute intervals, provide information about water mixing within the lake which can be used to predict the development of algal blooms. In [145], a flood monitoring and detection system has been developed. The system takes into account information such as humidity, temperature, water level, and amount of rainfall as flood indicators. The sensor deployed in the sensor field senses the information and transmits it to the remote station where, on crossing the threshold, the vicinities are notified through Short Message Service (SMS). The system currently covers 15 flood prone regions in Uyo metropolis in Akwa Ibom state, Nigeria. The authors in [166] have developed an application to monitor the quality of pool water for trout farms. For the growth of trout in a farm/pool, various parameters were monitored such as chemical oxygen demand, ammonium nitrogen (NH3-N), pH, and electrical conductivity (EC). The parameters were monitored for 270 days between August 2011 and April 2012. An algorithm was proposed by the authors which can display the information of the input and output of all the four pools. The comparison was made using fuzzy logic for evaluating the sensed data and notifying in the case of any critical state whenever the parameters surpass the threshold values.

1.3.3/ Industrial Monitoring

The use of PSNs for industrial applications has attracted much attention from both academic and industrial sectors. It enables a continuous monitoring, controlling, and analyzing for the industrial processes and contributes significantly to find the best operations performance. Industrial PSN applications cover the problems of air pollution, structural condition monitoring, production performance monitoring, evaluation and improvement. For instance, continuous monitoring of pressures eliminates the need for daily visits to the wellhead to manually record gauge readings.

RealFusion project [96] is a good example of PSNs applications in the industrial environment. RealFusion develops two PSNs: the objective of the first one is to monitor, for one week, the amount of bulk substances in five silos in a factory. For each silo, sensors collect the amount of bulk substances once every 8 seconds then, they send the collected data with their sensing collection date to the sink node located at the factory main office. The second network has implemented with 9 sensors in order to monitor the environment conditions, such as temperature and humidity, in an industrial warehouse for about 10 weeks. At each period of 16 minutes, each sensor collects data then send them to an access points which, in its turn, forwards them to the server located in the main office of the company. Another application for PSN in the industrial environment is developed in order to optimize the oil reservoir production [169]. The developed network system can ensure the real time monitoring of the oil characteristics, such as temperature and humidity. Data gathered by the sensors are processed and forwarded to a sink server for studying and analyzing purposes in a latter time. The authors in [141] develop a continuous real-time PSN for machinery condition-based maintenance (CBM) in small machinery spaces using commercially available products. Their proposed monitoring system has been tested in Heating & Air Conditioning Plant in Automation and Robotics Research Institute in University of Texas. In [162], the authors deployed a wireless data acquisition system that is used for damage detection on the building. Their proposed system continuously collects structural response data from a multi-hop network of sensor nodes, displays and stores the data in the base station. Finally, the noise pollution in urban areas using PSN has been studied in [128]. The authors present a prototype of a platform for collection and logging of the outdoor noise pollution measurements. These measurements can be used for the analysis of pollution effect on manpower productivity and social behavior.

1.3.4/ Healthcare Monitoring

Continuous monitoring is becoming a requirement for offering a better healthcare to an increasing number of patients whether they are in hospital or at home. Nowadays, periodic sensor networks play a central role in healthcare applications as an efficient and low cost solution. PSNs monitor the physiological conditions of (remote) patient and transmit the data periodically to some remote location without human intervention. A doctor can interpret these sensor readings to assess a patient's condition or to provide urgent treatment while an emergency occurs. PSNs in healthcare applications include the monitoring of (remote) patient vital signs (body temperature, heartbeat, blood pressure, oxygen saturation, ...), aged care, mass casualty disaster monitoring, diseases detection (asthma, stress, cancer, parkinson, alzheimer...), preventing medical accidents, etc.

In [84], the authors developed PSNs on the purpose to monitor the activities of the elderly persons. By continuously monitoring their activities, the proposed network analyzes the sensed data, detects the anomalies activities and alerts when necessary. In order to detect the elderly's activities, a set of sensor tags have been attached to items in the user's home and on the user's hand. When the tag on the hand picks items in the home, the information is logged automatically on an electronic daily activity forms. Thus, by analyzing such forms, the health professionals can notice any anomalies or deviations in the activities of the observed elderly. HipGuard [64] is another interesting periodic application of sensor networks in healthcare. HipGuard is a detection system dedicated to monitor hip replacement patients in their homes from eight to twelve weeks after the operation. The proposed system is mainly based on a pair of pants with 7 sensors placed on the pants in order to collect data of the operated hip and alert the user when the load on the operated hip meet a defined limits. In [58], the authors have developed a network with 45 sensors based location system in NTUH-BH to automatically track the elder's daily mobility. The objective of their project is to monitor the elderly' physical and mental health which is slowly declining due to multiple chronic illnesses. The sensors collected location traces and investigated the daily and long-term mobility of four volunteering elders for eight months. Based on the network observations, they concluded that long term location tracking allows discovery of the moving patterns and in turn making early detection of the elders' physical or mental problems possible.

1.4/ Network Architecture

After being deployed in the field of interest, sensor nodes organize themselves in the network with the sink node. The architecture of the sensor network plays a vital role in WSNs since it has a significant impact on energy consumption, capacity and reliability of the network. The objective of any WSN is to manage the data in an energy efficient manner in order to maintain the lifetime of the sensor nodes as long as possible. Indeed, the performance of a data management strategy depends on the network architecture, which is a key criteria in WSNs. As sensor nodes are energy constrained, it is inefficient for sensor nodes to transmit data directly (e.g. in single-hop) to the sink for two reasons. The first reason is that in large networks, some nodes may reside in areas which are far away from the sink, thus, direct communication is completely impossible in single-hop. The second reason is that even when direct communication is possible, it is better to send packets using multi-hop routing towards the sink, as power consumption is proportional, in one hand, to the transmission distance and, on the other hand, to the amount of data transmitted. Thus, multi-hop routing consumes less energy than direct communication due to ability to reduce transmission distance by transmitting data to the neighbor nodes, and to reduce amount of data transmitted by performing aggregation processing at intermediate nodes [138]. Therefore, several number of schemes for WSNs have been proposed by researchers based on multi-hop communication between sensor nodes and sink [93]. However, clustering and tree based schemes are still the most used architecture in this regard.

In the tree-based architecture, the deployed sensor nodes construct a logical tree whereas the sink serves as a root for this tree. After tree building is done, data are transmitted hop by hop from leaf nodes to the root. In this scheme, data are processed and aggregated at intermediate nodes, e.g. parent nodes, before sending them to the root. Figure 1.2 shows an example of WSN based on tree architecture. However, tree-based architecture has several disadvantages:

- Tree's construction is very costly in term of time consumption.
- Sensor nodes close to the sink will die out first, as the aggregation process is more complex in these nodes compared to other nodes in the tree.

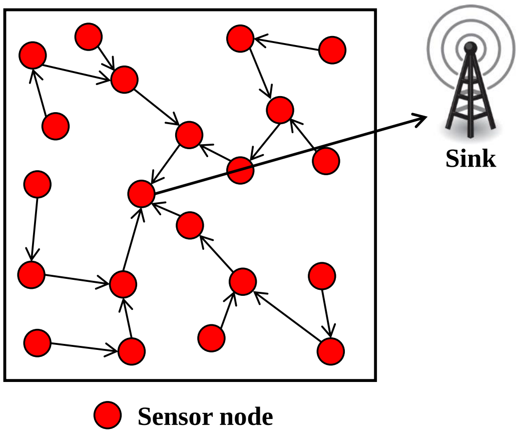


Figure 1.2: Tree-based network architecture.

- It is sensitive to node failures, especially nodes close to the root. If a parent node fails, then its sub-tree becomes disconnected until the network is reorganized.
- It is sensitive to packet loss: when a packet is lost at a given level of the tree, all the aggregated data generated by the related sub-tree are lost.
- The delay is high when sending data from leaf nodes to the root, especially in dense networks.

Consequently, cluster-based scheme has been proposed as an efficient way to pass most drawbacks of tree scheme. In addition, cluster scheme can efficiently manage the power consumption and achieve the network scalability objective [15, 97]. In such architecture, the whole network is divided into several clusters, each cluster has a cluster-head (CH). Each CH is responsible for managing its cluster. In Figure 1.3, we show an illustrative example for a cluster-based sensor network where sensed data reach their destination (the sink) by travelling via CHs.

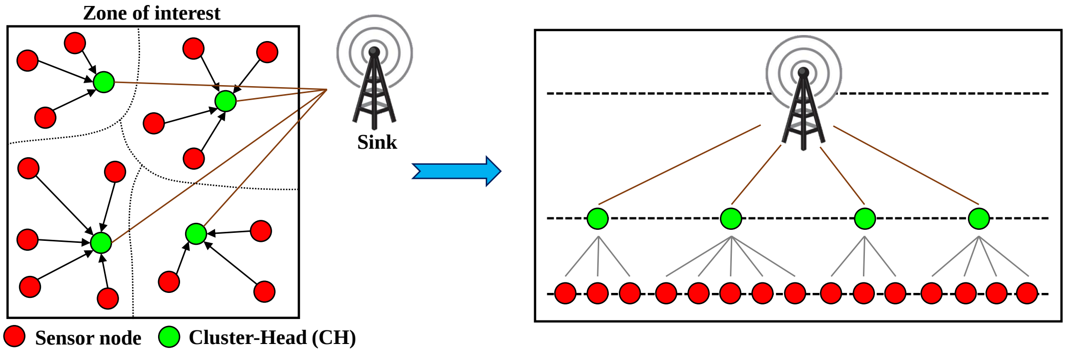


Figure 1.3: Cluster-based network architecture.

In the literature, the challenges related to the clustering architecture have been treated by a great number of researchers: some aim at forming and maintaining the clustered networks while optimizing cluster size [154, 171, 45, 10], others try to select the Cluster-Head (CH) or to change the entire cluster hierarchies periodically [154, 167, 2, 100], others are interested in communication among nodes and among clusters [109, 13, 83] or in cluster joining [68]. Hence, in this thesis, we do not treat such challenges neither the forming of the clusters. We assume that the network is already clustered using the appropriate clustering scheme. Then, our objective is to study the correlation between data generated in each cluster. Therefore, we consider the following assumptions:

- CHs are defined during the deployment phase.
- Data transmission between member nodes and their appropriate CH or between CHs and the sink is based on single-hop communication.
- Sensor nodes collect data in a periodic manner. Subsequently, each member node sends periodically (at each period p) its data to the appropriate CH, which in its turn sends it to the sink.
- Sensor nodes sense environment at a fixed rate where each of one takes τ measures at each period.
- There is no constraint associated to the sink node.

1.5/ Periodic Sensor Network Challenges

PSNs are characterized by the acquisition of data from large number of sensor nodes, distributed over vast areas, before being forwarded to the sink in a periodic basis. Such type of networks provide several challenges for research community. In this section, we describe the main challenges involved by PSNs (some of them can be common to the traditional WSNs).

(I) Deployment: The deployment of the sensors can be considered as the first challenge for sensors networks since it is the first operation (phase) in the life cycle of the network. Depending on the needs of applications, there are two ways to sensors deployment: deterministic or randomly. In the first way, network can be placed sensor by sensor in a deterministic manner by a human or a robot. In the second way, sensor nodes can be deployed randomly from a plane or a roket for example. However, the manual deployment is impossible in many applications. In addition, even when the application permits deterministic deployment, the random deployment is adopted in the majority of scenarios because of practical reasons such as cost and time. On the other hand, the random deployment cannot provide a uniform distribution on the region of interest, triggering new problems in sensor networks such as location [75] and network connectivity [174].

(II) Network Lifetime /Energy Consumption: The major challenge that faces PSNs is improving the lifetime of the network; in other words ensuring a long-time monitoring for a given area [95, 18]. Indeed, the network lifetime is highly related to the power consumption of sensors batteries as they are the only source of energy. In addition, it is difficult and cost ineffective to recharge the sensors batteries in most cases, because nodes may be deployed in a hostile or unpractical environment. Therefore, one of the major objectives today is to efficiently manage the energy of the sensor in order to increase the network lifetime. Hence, a lot of solutions, i.e. hardware and software, have been proposed to optimize the energy consumption thus, to maximize the lifetime of the sensor network [90]. The hardware solutions seek to produce components dedicated to wireless sensor networks with the best power/watt consumed, such as fuel batteries. Otherwise, the goal of the software solutions is to develop distributed algorithms to minimize energy consumption, such as scheduling sensor strategies or data aggregation algorithms.

(III) Density/Scalability: The sensor network is usually distributed redundantly in hundreds or thousands over a deployment zone of interest. Such redundant volume is required to ensure the cooperative sensors readings reliability. Therefore, the explosive growth of the data volume generated in PSNs has led to one of the most challenging research issues of the big data era. Such amounts of data provide several problems in PSNs: a high level of data redundancy, packets collision in the network, increasing energy consumption, complicated mission for decision makers. Therefore, data collection and aggregation techniques have received a great attention in PSNs in order to reduce the amount of generated data [125, 143].

(IV) Routing: The task of finding and maintaining routes between the sensors and the sink is very challenging in PSNs. The design of routing protocols face several constraints, dependent on the capabilities of nodes (limited transmission range, limited energy capacity, limited processing and storage) and on the inherent features of network (self-configurable, sensors locations, identifications of nodes, faults tolerance, topological changes). Therefore, several routing protocols have been proposed in the literature for PSNs in order to discover the routes in the network and to ensure a reliable multi-hop communication under these constraints. To minimize energy consumption, the majority of such protocols employs some well-known routing tactics, e.g. in-network processing, data aggregation and clustering [73, 103, 3].

(V) Coverage: Preserving maximal coverage of the region of interest is another important challenge in PSNs . After they have been deployed, sensor nodes must ensure a satisfactory sensing coverage during each period of the network lifetime. Otherwise, some applications in PSNs permit flexibility regarding coverage of the network such as environmental monitoring, while another applications are considered critical where the area of interest should be always full covered such as industrial or military surveillance. Hence, scheduling sensors strategies have been proven to be an efficient mechanism in PSNs when coverage problem is considered [99, 111]. The basic idea is to select a set of sensor nodes in order to monitor the target field without affecting the application requirements.

(VI) Synchronization: Time synchronization is a significant and costly challenge in sensor networks. Many sensor network applications require synchronization of the local clocks of the nodes. For example, in tracking and vehicular surveillance, the estimated trajectory of the tracked object could differ significantly from the actual one without an accurate time synchronization scheme. Obviously, there is no specific time synchronization scheme available to achieve higher order of accuracy with greater scalability independent of topology and application [122]. However, the single-hop cluster-based PSN can be considered is the most important network that achieves time synchronization between distributed sensor nodes. This is because, data generated by sensors will be transmitted to the CH at the same time (at the end of each period) in one hop communication thus, an accurate time information can be approximately guaranteed. Indeed, the loss or the delayed of packets constitutes a real problem for time synchronization in PSNs. This is because the CH must wait, at each period, all data packets coming from its member nodes before sending them to the sink. Consequently, any loss or delayed of packet can change the time synchronization of data at CH and then, at the sink.

(VII) Security: Since the deployment of sensor nodes in an unattended environment makes the networks vulnerable to a variety of potential attacks, security issue is another challenge for PSNs. Such security takes more attention in some applications such as battlefield areas and monitoring of critical infrastructure. Indeed, sensors scattered in open areas must be able to keep private the information that collect. However, the inherent power and memory limitations of sensor nodes makes conventional security solutions unfeasible [110]. On the other hand, the physical security of the sensors should be always ensured during the network lifetime. Because of the widespread placement in an often non-secure area, sensor nodes are subject to be attacked by the invaders or more simply the animals. Hence, it is important to develop a complete framework that takes into account the physical/information security of the sensors and the requirements of the monitored application.

(VIII) Data Survivability: In periodic sensor networks, each sensor node needs to collect and store data during each period before sending them toward the sink. Indeed, a high failure rates lead to significant loss of data thus, data survivability becomes a real challenge in PSNs for maintaining network reliability. There are two existing strategies to achieve survivability of data: replication and coding. Although replication based solutions ca ensure a high level of data reliability, coding based solutions remain more suitable for periodic sensor networks. Replication often requires lots of storage on every node at each period. In addition, replication based approaches also need to keep track of where different data exists, resulting in complicated data gathering protocols [5]. However, coding based solutions have been shown to greatly reduce storage requirements as well as simplify data gathering mechanisms [40]. Finally, taking attention to the limited resources of nodes, data survivability represents a quantitative design requirement for data resilience in PSNs in order to prevent loss of data in the network in case of data failure.

1.6/ Data Management Issues

Speaking about more than hundreds, and sometimes thousands, of sensors, which are randomly distributed for periodic monitoring, makes PSNs one of the big data producers. This makes data management in such networks very complex. First, data collected in PSNs are usually redundant and correlated which makes the analysis of this data a complicated mission for the decision makers. Then, the transmission of such amount of data is very expensive in terms of energy [119, 11, 16, 49]. Therefore, designing an energy efficient data management strategies for PSN are considered as important issue to extend its lifetime. In this section, we describe different issues related to the data management in periodic sensor networks (PSNs).

1.6.1/ Data Collection

Sensors are typically deployed to collect data about the surrounding environment and transmit them at a fixed periodicity to the sink node. Hence, data collection can be considered is one of the fundamental operations in PSNs. Mostly, readings data collected by sensor nodes are highly dependent on the monitored condition [76]; when the monitored condition slows down or speeds up, the readings collected by each sensor, within each period and among successive periods, are more redundant. Therefore, it is important to monitor carefully the amount of data to collect and send, and the frequency at which it is collected and sent, while preserving the quality of service expected by the application. Hence, adaptive sampling approach to periodic data collection constitutes a fundamental mechanism for energy optimization and data reduction in PSNs. The main objective of such approach is to allow each sensor node to adapt its sampling rate according to the dynamics of the monitored condition, in order to prevent collecting redundant measures and saving energy of the sensor. As a result, adaptive sensor sampling rate has been received, during the last years, a great deal of research attention due to its capability of enhancing network lifetime.

1.6.2/ Data Aggregation/In-Network Data Aggregation

In PSNs, the huge number of collected and transmitted data from sensors leads to consume most of the available energy of sensor nodes. Therefore, aggregating data is an essential process in order to reduce the amount of data transmission, and to improve the energy consumption of the network [18, 120]. Data aggregation process has as first goal eliminating redundancy in data collected from sensor nodes, thus sending only the useful information to the sink. Mostly, the performance of a data aggregation technique depends on the network architecture, which is a key criteria in sensor networks. Recently, combining the node clustering and the data aggregation have been proven as very efficient framework that organizes data traffic and reduce in-network redundancies while improving scalability and energy consumption [73, 170, 139, 74, 33]. While nodes clustering makes a network look smaller by reducing transmission hops between the nodes and the sink, the data aggregation can to minimize the number of transmissions between them. In such framework, CHs can perform, sometimes, in-network data aggregation processing in order to aggregate data received from neighboring sensor nodes then send aggregated data to the sink; due to the random and the dense network deployment, nodes may have overlapping sensing ranges, such that events can be detected by neighboring sensor nodes providing a redundancy in sensed data.

1.6.3/ Data Correlation

In PSNs, the densely deployment and the dynamic phenomenon provide strong correlation between sensor nodes. This correlation is typically spatio-temporal. Spatial correlation usually exists among the readings of close sensor nodes. Indeed, spatially proximal sensor observations are highly correlated with the degree of correlation increasing with decreasing inter-node distance. Furthermore, readings from a senor node can be predicted from that of its neighboring sensor nodes with high confidence. Therefore, neighboring nodes with a certain level of spatial correlation can be turned off and only a subset of the sensor nodes are turned on for sampling and data transmission, in order to save energy. On the other side, temporal correlation in PSNs usually exists in two cases: in the time series from a single sensor node and between neighboring nodes. In the first case, it means that the future readings of a sensor node can be predicted based on the previous readings of the same node. In this case, the degree of correlation between consecutive sensor readings may vary according to the temporal variation characteristics of the phenomenon [151]. In the second case (inter-node temporal correlation), it means that during the same period neighboring nodes generate very similar and correlated data sets. It is worth noting that spatially correlated nodes can generate dissimilar sensor data and turning off one sensor can lead to erroneous analysis. Therefore, it is important to exploit the inter-nodes temporal correlation beside the spatial correlation in order to eliminate the redundancy and improve the network's lifetime. As a result, energy consumption can be reduced and long term data collection can be ensured by exploiting inter-nodes correlations.

1.6.4/ Data Latency

Real-time delivery data is an important factor in periodic sensor networks in order to take important decisions as fast time as possible. Some applications, such as natural disasters (volcano, seismic, tsunami, etc.) and critical infrastructure monitoring (nuclear, biological, etc.), require periodic and fast data delivery to the end user over long periods, or millions of people will be attacked. Therefore, minimizing latency is essential for periodic applications where the collected data should be transmitted to the sink in a minimum amount of time. In addition, reducing data latency is a major issue for data aggregation where the used algorithms must achieve the minimum delay when eliminating redundancy [17]. Therefore, latency is a key element to evaluate the performance of any in-network data aggregation technique.

1.6.5/ Data Accuracy/Information Integrity

The accuracy of the sensed data is one of the key criterions in PSNs because it affects the decision of the end user. Data loss in sensor networks is common and has its special patterns due to noise, collision, unreliable link and unexpected damage [69]. Such reasons are sufficient sometimes to make the network out of service when missing data becomes large. Most of the work done today is based upon the fact that the sink node is responsible for estimating the data accuracy for physically sensed data by sensor nodes [31]. Thus, it must reconstruct the whole data based on the received one and the correlation between sensors in the network. In addition, data aggregation performed in PSNs can decrease the accuracy of the collected information when eliminating the redundancy from the raw data. Mostly, the CH aggregates data coming from its member nodes before sending them to the sink which leads to loss some data according to the aggregation process. Therefore, data aggregation must be highly energy efficient without affecting the integrity/fidelity of the information.

1.7/ Conclusion

In this chapter, we have introduced periodic sensor networks in which data are collected and sent on a periodic basis to the sink. Then, we presented some examples of periodic applications including environmental, underwater, industrial and healthcare fields. The clustering scheme is also described in this chapter as an efficient architecture for the periodic sensor networks. After that, we have presented challenges that face PSNs while highlighting the energy consumption as the primary challenge to be optimized in order to increase the network lifetime. Finally, we have described the data management as a big challenge for PSNs while highlighting the data collection, aggregation and correlation in such networks. Next, we present in more details our proposed techniques for data management in periodic sensor networks (PSNs).

2

Adaptive Real-Time Data Collection Model

Massive data collected by the sensors besides the limited battery power are the main limitations imposed by the periodic sensor networks (PSNs). In this chapter, we propose an efficient adaptive model of data collection for PSN in order to reduce the huge amount of collected data thus, increasing the network lifetime. The main idea behind this approach is to allow each sensor node to adapt its sampling rate to the physical changing dynamics. By this way, the oversampling can be minimized and the power efficiency of the overall network system can be further improved. The proposed method is based on the dependence variance of readings while taking into account the residual energy of the sensor. We study three statistical tests (Fisher, Tukey and Bartlett) based on one-way ANOVA model. Then, we use an existing multiple level activity model that uses behavior functions modeled by modified Bezier curves to define application classes and allow each sensor to adapt its sampling rate.

2.1/ Introduction

Wireless Sensor Networks (WSNs) have become a highly active research area due to their increasing potential impact on the quality of people's lives. A main application domain where wireless sensor networks are broadly used is environmental data collection and monitoring, where certain conditions or processes need to be monitored periodically, such as the temperature in a conditioned space or pressure in a process pipeline [76]. In such applications, data generated across numerous sensors can produce a significant portion of the big data. Hence, periodic data collection provides two major challenges. First, the network should have a lifetime long enough to fulfill the application requirements. Second, massive and heterogeneous data collected from networks make data management more complex. Researchers' strategies are often targeted to minimize the amount of data collected/communicated by the network without considerable loss in fidelity/accuracy. The goal of this reduction is first to increase the network lifetime, by optimizing energy consumption of the limited battery for each sensor node, and then to help in analyzing data and making decision. Subsequently in periodic monitoring, the dynamics of the monitored condition or process can slow down or speed up; if the sensor node can adapt its sampling rates to the changing dynamics of the condition or process, over-sampling can be minimised and power efficiency of the overall network system can be further improved [76]. Therefore, in order to keep the network operating for long time, adaptive sampling approach to periodic data collection constitutes a fundamental mechanism for energy optimization and data reduction.

In this chapter, we propose an efficient adaptive sampling approach based on the dependence of the conditional variance on readings, e.g. one-way ANOVA model and statistical tests, that varie over time. We study three different statistical tests (Fisher, Tukey and Bartlett) based on the one-way ANOVA Model while taking into consideration the residual energy of each node. Indeed, the ANOVA model provides a statistical test of whether or not the means of several independent groups are all equal. It has been proved as an effective method to classify objects (or data) into groups whereas statistical tests are used to indicate which groups are significantly different [60]. As a common method in statistical inference, ANOVA has many applications in agricultural, biological, and engineering sciences [124]. Then, we use an existing multiple level adaptive model that takes into account the application criticality. It defines dynamically multiple levels of sampling rate corresponding to how many samples are captured per unit of time (or period). It uses behavior functions modeled by modified Bezier curves to define application classes and allow for adaptive sampling rate [76]. Simulation results are presented to validate the performance of the proposed approach.

The remainder of this chapter is organized as follows. Section 2.2 gives an overview about the existing techniques for adapting sensor sampling rate in data collection in sensor networks. Section 2.3 introduces the ANOVA model used in [76] with different statistical tests for adapting sensor sampling rate based on the variance study. Section 2.4 describes how to integrate the residual energy to allow each node to compute its sampling rate. Experimental results are exposed in Section 2.5. Finally, we conclude our chapter in Section 2.6.

2.2/ Data Collection: A Background

Although there has been a large number of works on data collection in sensor networks, only a fairly small number explicitly deals with adaptive sensor sampling approach. The main goal of an adaptive sampling approach is to make the rate of sensing dynamic and adaptable; if the sensor node can adapt its sampling rates to the changing dynamics of the condition or process, over-sampling can be minimized and the computational load at the sink will be more flexible.

Adapting sampling rate for the sensor is not a new concept [46, 65, 152, 158, 8, 147, 173]. In [46], the authors propose an Adaptive Sampling Approach to Data Collection (ASAP) which splits the network into clusters. A cluster formation phase is performed to elect cluster heads and select which nodes belong to a given cluster. The metrics used to group nodes within the same cluster include the similarity of sensor readings and the hop count. Then, not all nodes in a cluster are required to sample the environment. A centralized adaptive method is proposed in [65], where the sampling rate is derived based on a Kalman filter. In this case, the sink establishes the sampling rate of nodes. The authors in [152] define a spatial Correlation based Collaborative MAC protocol (CC-MAC) that regulates sensor node transmissions so as to minimize the number of reporting nodes while achieving the desired level of distortion. A temporal correlation of data is used in [8], where the authors propose an adaptive sampling scheme suitable to snow monitoring for avalanche forecast. A TA-PDC-MAC protocol is proposed in [147], a traffic adaptive periodic data collection MAC which is designed in a TDMA fashion. This work is designed in the way that it assigns the time slots for nodes activity due to their sampling rates in a collision avoidance manner. The authors in [173] propose an adaptive sampling which basically consists in activating the appropriate number of sensor nodes to achieve a target error level, depending on spatial correlation and activity.

Recently, researchers proposed various methods of adapting sampling rate in sensors [94, 127, 78, 159, 102]. In such techniques, the sensor adapts its sampling rate based on the correlation between sensed data. The authors in [94] propose an energy-efficient adaptive sampling mechanism which employs spatio-temporal correlation among sensor nodes and their readings. The main idea is to carefully select a dynamically changing subset of sensor nodes to sample and transmit their data. In [127], an Efficient Data Redundancy Reduction (EDRR) scheme is proposed. EDRR integrates conjugative sleep scheduler scheme and basically utilizes Differential Pulse Code Modulation (DPCM) technique to reduce data redundancy over the network. In [78], the authors develop an automatic auto regressive-integrated moving average modeling-based data aggregation scheme in WSNs. The developed scheme can decrease the number of transmitted data values between sensor nodes and aggregators by using time series prediction model. A machine learning architecture for context awareness is used in [159] which is designed to balance the sampling rates (and hence energy consumption) of individual sensors with the significance of the input from that sensor.

Adaptive sampling techniques are very promising, because of their efficiency to optimize energy consumption and the network overload. However, most of the previous proposed solutions are implemented in a centralized manner that requires huge computations and communications. Other existing methods are limited to only space correlation and are based on grouping nodes into clusters. In addition, all proposed methods consider that all applications have the same criticality level which it is not always true since the risk level can be changed from application to another. Recently, the authors in [76] proposed an adaptive sampling approach based on the dependence of the conditional variance on measurements calculated with Fisher test. It allows each sensor node to adapt its sampling rate to the physical changing dynamics. In this chapter, we study and compare three different tests: Fisher, Barttlett, and Tuckey using the one-way ANOVA model. We show by experimental results that Fisher test is not the best choice for PSN in terms of energy saving. Furthermore, previous works consider only environmental variations to adapt the sampling rate and none of them takes into account the intrinsic constraints of the nodes such as the residual energy level. The approach presented in this chapter allows each node to adapt its sampling rate depending on its residual energy level.

2.3/ Adapting Sensor Sampling Frequency

In this section, we present our proposal for an adaptive model to calculate the sampling frequency of each sensor node. First, we present the model of adaptive sampling rate proposed [76] which is based on the variance study. In such study, the one-way ANOVA model has been used to determine whether there are any significant differences between the means of different data sets collected in successive periods. Then, we propose to extend the one-way ANOVA model to other statistical tests (Tukey and Bartlett) in order to compare the factors of the total deviation.

2.3.1/ Data Variance Study based on ANOVA Model

In this section, we recall the one-way ANOVA model used in [76] to calculate the sampling frequency of each sensor node. In [76], ANOVA model is used to test the total variation (TV) of data generated by a sensor node S_i in J periods. TV can be calculated based on the variation within period (VWP) and the variation between period (VBP) as follows:

$$TV = VWP + VBP \Rightarrow$$

$$\sum_{j=1}^{J}\sum_{i=1}^{n_j}(r_{ij} - \overline{Y})^2 = \sum_{j=1}^{J}\sum_{i=1}^{n_j}(r_{ij} - \overline{Y_j})^2 + \sum_{j=1}^{J} n_j \times (\overline{Y_j} - \overline{Y})^2 \tag{2.1}$$

Where:

- r_{ij} : i^{th} reading taken by the sensor S_i in the j^{th} period,
- n_j : Number of readings in j^{th} period,
- n_J : Total number of readings in all J periods,
- $\overline{Y_j}$: Mean of data in the j^{th} period,
- $\overline{Y}$: Mean of data in all J periods.

Consequently, if the variance caused by the interaction between the readings is much larger to that appeared within each period, then the means of periods are not the same.

2.3.2/ Statistical Tests

In statistics, a lot of statistical tests have been proposed by different researchers [101, 1]. The objective of a statistical test is to verify if the variation between sets of data is above a certain threshold. In this section, we are interested in three statistical tests: Fisher, Tukey and Bartlett. Bartlett's Test seems to be the most uniformly powerful test for the homogeneity of variances problem in the case that the data are normal. However, it has a serious weakness if the normality assumption is not met. Consequently, in such case, we must adopt other tests like Fisher and Tukey. For these reasons, we compared these different tests. In the next, we show how we can apply these tests to allow each sensor node to adapt its sampling rate. Let us start by Fisher test studied in [76].

2.3.2.1/ Fisher Test

Fisher test is a statistical hypothesis test for verifying the equality of two variances by taking the ratio of the two variances and ensuring that this ratio does not exceed a certain theoretical value (that we can find in Fisher's table). Let:

$$F = \frac{VBP/J - 1}{VWP/n_J - J} \tag{2.2}$$

Thus, the decision is based on the next:

- if $F > F_t = F_{1-\alpha}(J-1, n_J - J)$ the variance between periods is significant with a false-rejection probability α.
- if $F \leq F_t$ the variance between periods is not significant thus the readings captured in the J periods are considered redundant.

Note that F_t is a threshold which can be searched in Fisher's table based on J, n_J and α values.

2.3.2.2/ Tukey Test

Tukey post-hoc test [52] is a method that is used to determine which periods among total periods have significant differences. This method calculates the difference between the means of the periods. Tukey's test values are numbers which act as a distance between the periods. It works by defining a value known as Honest Significant Difference (HSD) as follows:

$$SS_{total} = (\sum_{i=1}^{n_1} r_{1i}^2 + \sum_{i=1}^{n_2} r_{2i}^2 + \cdots + \sum_{i=1}^{n_J} r_{Ji}^2) - \frac{(\sum_{i=1}^{n_1} r_{1i} + \sum_{i=1}^{n_2} r_{2i} + \cdots + \sum_{i=1}^{n_J} r_{Ji})^2}{n_J} \tag{2.3}$$

$$SS_{among} = (\frac{(\sum_{i=1}^{n_1} r_{1i})^2}{n_1} + \frac{(\sum_{i=1}^{n_2} r_{2i})^2}{n_2} + \cdots + \frac{(\sum_{i=1}^{n_J} r_{Ji})^2}{n_J}) - \frac{(\sum_{i=1}^{n_1} r_{1i} + \sum_{i=1}^{n_2} r_{2i} + \cdots + \sum_{i=1}^{n_J} r_{Ji})^2}{n_J} \tag{2.4}$$

$$SS_{within} = SS_{total} - SS_{among}; \quad df_{among} = J - 1; \quad df_{within} = n_J - J$$

$$MS_{among} = \frac{SS_{among}}{df_{among}}; \quad MS_{within} = \frac{SS_{within}}{df_{within}}; \quad F = \frac{MS_{among}}{MS_{within}}$$

Where:

- SS_{within} : Sum of Squares within periods,
- SS_{among} : Sum of Squares between periods,
- MS_{within} : Mean Squares within periods,
- MS_{among} : Mean of squares between periods,
- J : Number of total periods,
- n_J : Total number of readings (all periods),
- n_j : Number of readings in j^{th} period.

The “Variance Between Periods” represents what is often called “explained variance” or “systematic variance”. We can think of this as variance that is due to the independent variable, the difference between the two periods. For example the difference between a reading in period one and a reading in period two would represent an explained variance. The “Variance Within Periods” represents what is often called “error variance”. This is the variance within periods, variance that is not due to the independent variable. For example, the difference between one reading in period 1 and another reading in the same period would represent error variance. Intuitively, it is important to understand that, at its heart, the analysis of variance and the F score it yields is a ratio of explained variance versus error. Therefore, when we calculate df_{among}, df_{within}, MS_{among}, and MS_{within}, and F we check if F is statistically significant on probability table with an appropriate degree of freedom $F_t = df(df_{among}, df_{within})$.

The decision is based on F and F_t:

- if $F > F_t$ the hypothesis is rejected with false-rejection probability α, and the variance between periods are significant.
- if $F \leq F_t$ the hypothesis is accepted.

2.3.2.3/ Bartlett Test

Bartlett test [123] is used to test if J periods are from data with equal variances. Equal variances across samples is called homogeneity of variances. Some statistical tests, for example the analysis of variance, assume that variances are equal across periods. The Bartlett test can be used to check that assumption. Bartlett's test is sensitive to departures from normality. That is, if the periods come from non-normal distributions then Bartlett's test may simply be testing for non-normality. Bartlett's test is used to test the null hypothesis, H0 that all J periods variances are equal against the alternative that at least two are different. If there are J periods with size n_j and variance σ_j^2 for each one then Bartlett's test is applied as follows:

$$F = \frac{(n_J - J)\ln(\sigma_p^2) - \sum_{j=1}^{J}(n_j - 1)\ln(\sigma_j^2)}{\lambda} \tag{2.5}$$

where :

$$n_J = \sum_{j=1}^{J} n_j; \quad \lambda = 1 + \frac{1}{3(J-1)}\Big(\sum_{j=1}^{J}(\frac{1}{n_j - 1}) - \frac{1}{n_J - J}\Big) \tag{2.6}$$

In the above:

- J is the total number of periods,
- n_j is the number of readings in the j^{th} period,
- σ_j^2 is the variance in the j^{th} period which can calculated as follows:

$$\sigma_j^2 = \frac{1}{n_j}\sum_{i=1}^{n_j}(r_{ij} - \overline{Y_j})^2$$

- σ_p^2 is the pooled variance, which is a weighted average of the period variances and it is defined as:

$$\sigma_p^2 = \frac{1}{n_J - J} \sum_{j=1}^{J} \sigma_j^2 \; (n_j - 1)$$

Bartlett's test has approximately a $(J-1)$ degrees of freedom. Thus the null hypothesis is rejected if $F > F_{J-1,\alpha}$ (where $F_{J-1,\alpha}$ is the upper tail critical value for the F_{J-1} distribution). To unify the notation of the three tets, we suppose that $F_t = F_{J-1,\alpha}$, thus the decision is based on the next:

- if $F > F_t$ the hypothesis is rejected with a false-rejection probability α, and the variance between periods are significant.
- if $F \leq F_t$ the hypothesis is accepted.

2.3.3/ Adaptation To Application Criticality

Since the applications have different criticality level, the authors in [76, 95] define the risk level of an application by r^0 which can take values between 0 and 1 representing the low and the high criticality level respectively. This criticality level is represented by a mathematical function called BV (**B**eha**V**ior) function.

Then, in order to model the BV function, they use the Bezier curve which is flexible and can plot easily a wide range of geometric curves. Therefore, the BV function curve can be drawn, using the Bezier curve, through three points $P_0(0,0)$ (original point), $P_1(b_x, b_y)$ (behavior point) and $P_2(h_x, h_y)$ (threshold point). In Figure 2.1, we show how the curvature of the Bezier curve can be built when changing the behavior point (P_1) coordinates. As illustrated, the curve frame is delimited by the original and the threshold points, e.g. P_0 and P_1 respectively, while the behavior point moves through the diagonal of the rectangle in order to control the application criticality. Thus, when varying r^0 between 0 and 1, P_1 will update its position based on the following function [76, 95]:

$$\begin{aligned} Cr : [0,1] &\longrightarrow [0,h_x] \times [0,h_y] \\ r^0 &\longrightarrow (b_x, b_y) \end{aligned}$$

$$Cr(r^0) = \begin{cases} b_x = -h_x \times r^0 + h_x \\ b_y = h_y \times r^0 \end{cases}$$

Subsequently, the BV function is defined based on the Bezier curve as folows:

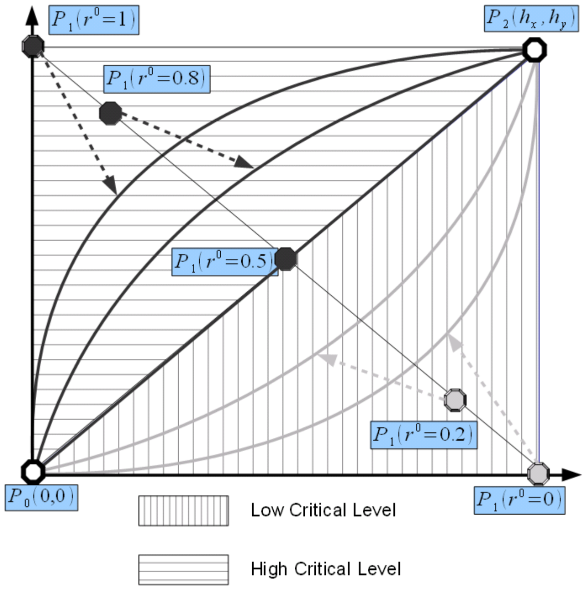


Figure 2.1: The Behavior curve functions ([76, 95]).

$$\begin{array}{rcl} BV : [0, h_x] & \longrightarrow & [0, h_y] \\ X & \longrightarrow & Y \end{array}$$

$$BV(X, h_x, r^0, h_y) = \begin{cases} \frac{(h_y-2b_y)}{4b_x^2}X^2 + \frac{b_y}{b_x}X & if\ (h_x - 2b_x = 0) \\ (h_y - 2b_y)(\propto (X))^2 + 2b_y \propto (X), & if\ (h_x - 2b_x \neq 0) \end{cases}$$

$$Where\ \ \propto (X) = \frac{-b_x + \sqrt{{b_x}^2 - 2b_x \times X + h_x \times X}}{h_x - 2b_x} \wedge \begin{cases} 0 \leq b_x \leq h_x \\ 0 \leq X \leq h_x \\ h_x > 0 \end{cases}$$

Adapting to Anova model and statistical tests, BV function takes, based on Bezier curve, four variables as input: the variance measures for a test F (replaces X), the test threshold F_t (replaces h_x), the risk level r^0 and the original sampling rate at the time of network deployment S_{max} (replaces h_y). Then, it returns the instantaneous sampling rate, S_t, calculated after each round.

2.4/ Adaptation to Residual Energy Level

In order to maintain proper operation of PSN and maximize its lifetime, it is important to use the network resources such as node energy efficiently. Since almost every sensor node is operated by limited battery power and since sensor nodes in PSN consume energy continuously by collecting and sending data packets, the longevity of a sensor node is inversely proportional to the number of data samples it collects. Ideally, sampling rate should be greater in nodes having higher energy levels relative to other nodes with low levels. Network operability will be prolonged if a critically energy deficient node can survive longer by reducing its sampling rate in function of its residual energy rather than operating on maximal sampling rate. Therefore, both application criticality and energy residual are critical metrics affecting the sampling rate adaptation and consequently the sensor lifetime.

2.4.1/ Analytical Study

In this section, we provide an analytical study to allow each sensor node to adapt its sampling frequency in function of its residual energy, the application criticality and the environmental dynamic changes. To do so, we consider that maximal sampling rate can not remain unchangeable and it must vary with the residual energy as follows:

$$S_{max} = \gamma \times q \times SMAX + \beta \times SMAX \tag{2.7}$$

where γ and β are two adaptive factors to adjust the impact of residual energy and environmental changes respectively. These factors depend on the variance readings F and the threshold F_t and their values are between 0 and 1. q is the probability for a sensor node to run if its residual energy is enough for the next period. $SMAX$ is the original sampling rate at the time of network deployment, and S_{max} is the instantaneous maximum sampling rate calculated after each round.

To compute the impact of the residual energy on the adaptive sampling, we must first find the values of γ and β. These values are complementary and calculated as:

$$\gamma = \begin{cases} 1 & \text{if } F < F_t \\ \frac{F_t}{F} & \text{if } F \geq F_t \end{cases} \tag{2.8}$$

$$\beta = 1 - \gamma \tag{2.9}$$

These values are found by the intuition that if the value of variance F is less than F_t (which means that there is no or very low changes in the environmental conditions) then the sampling rate must be calculated while considering only the residual energy as criteria ($\gamma = 1$). On the other hand, if F is greater than a threshold F_t we must adapt the sampling rate but in the same time without loosing critical information. Therefore, we consider that the value of γ must be proportional to the environmental conditions variations ($\frac{F_t}{F}$) and β increases when the variance F increases.

We consider that initially each node has E_0 energy and after each round r its remaining energy is E_r, then the probability q is computed in function of E_0 and E_r as follows:

$$q = \begin{cases} \ln(\frac{E_r \times e}{E_0} + 1) & \text{if} \quad 0 < E_r < E_0 - \frac{E_0}{e}, \\ 1 & \text{if} \quad E_r \geq E_0 - \frac{E_0}{e}. \end{cases} \tag{2.10}$$

This equation shows that, when the residual energy is large enough then q remains equal to 1, otherwise q is computed in function of E_r. In Figure 2.2, we show a distribution curve of q for $E_0 = 50$. Next, we present the adaptive algorithm while taking into account the residual energy.

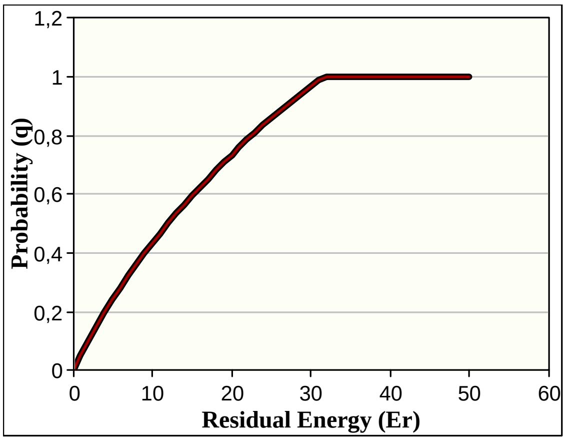


Figure 2.2: The Behavior curve of q for $E_0 = 50$.

2.4.2/ Adapting Algorithm

In this section, we present the adaptive sampling rate algorithm while taking into account the residual energy of the sensor (Algorithm 1). For each round, each node decides to increase or decrease its sampling rate according to the variance condition and the application risk and its remaining energy. While the energy is always positive, each node calculates the parameters F, F_t, γ, β and p and uses the BV function in order to find its new sampling rate.

2.5/ Experimental Results

In this section, we discuss the experimental results including the description of the chosen parameters and the adopted sensor network scenario. Two sets of experiments are presented to evaluate the effectiveness of our method. First we will show the results obtained while studying the variance and the application risk and secondly we provide results while taking the residual energy parameter into account for sampling rate adaptation. To verify our suggested approach, we conducted multiple series of experiments using a custom Java based simulator. The objective of these experiments is to confirm that our adaptive data collection technique can successfully achieve desirable results for energy conservation and data reduction in PSNs. Therefore, in our experiments we used real readings collected from 46 sensor nodes deployed in the Intel Berkeley Research Lab [88]. Mica2Dot sensors with weather boards collected timestamped topology information, along with humidity, temperature, light and voltage values once every 31 seconds.

For the sake of simplicity, in this study we are interested in one field of sensor readings: the humidity[1]. We performed several runs of the algorithms. In each experimental run, each node reads periodically real readings and adapts its sampling rate after each round according to its BV function. We evaluated the performance of the algorithm using the following parameters: **a)** The number of periods per round, P; **b)**The application criticality level, r^0; and **c)** The false-rejection probability, α. The period is fixed to 15 minutes, $SMAX$ to 15 readings and the initial energy (E_0) of the node is fixed to 50 units of energy.

[1] the others are done by the same manner.

Algorithm 1: Adaptive Sampling Rate Algorithm Based on Variance and Residual Energy

Data: P (1 round = P periods), $SMAX$ (maximun sampling speed), E_0 (initial energy), S_{max} (maximum sampling speed calculate)

Result: S_t (instantaneous sampling speed)

```
$S_t \leftarrow SMAX$;
while $E_r > 0$ do
    if $E_r \geq E_0 - \frac{E_0}{e}$ then
        $q \leftarrow 1$;
    end
    else
        $q \leftarrow \ln(\frac{E_r \times e}{E_0} + 1)$;
    end
    for $i = 1 \rightarrow P$ do
        takes readings at $S_t$ speed;
    end
    for each round do
        compute $F$;
        find $F_t$;
        if $F < F_t$ then
            $S_t \leftarrow BV(F, F_t, r^0, S_{max})$;
            $\gamma = 1$;
            $\beta = 1 - \gamma$;
        end
        else
            $S_t \leftarrow S_{max}$;
            $\gamma = \frac{F_t}{F}$;
            $\beta = 1 - \gamma$;
        end
    end
    $S_{max} = \gamma \times q \times SMAX + \beta \times SMAX$;
end
```

We employ three metrics in our simulations:

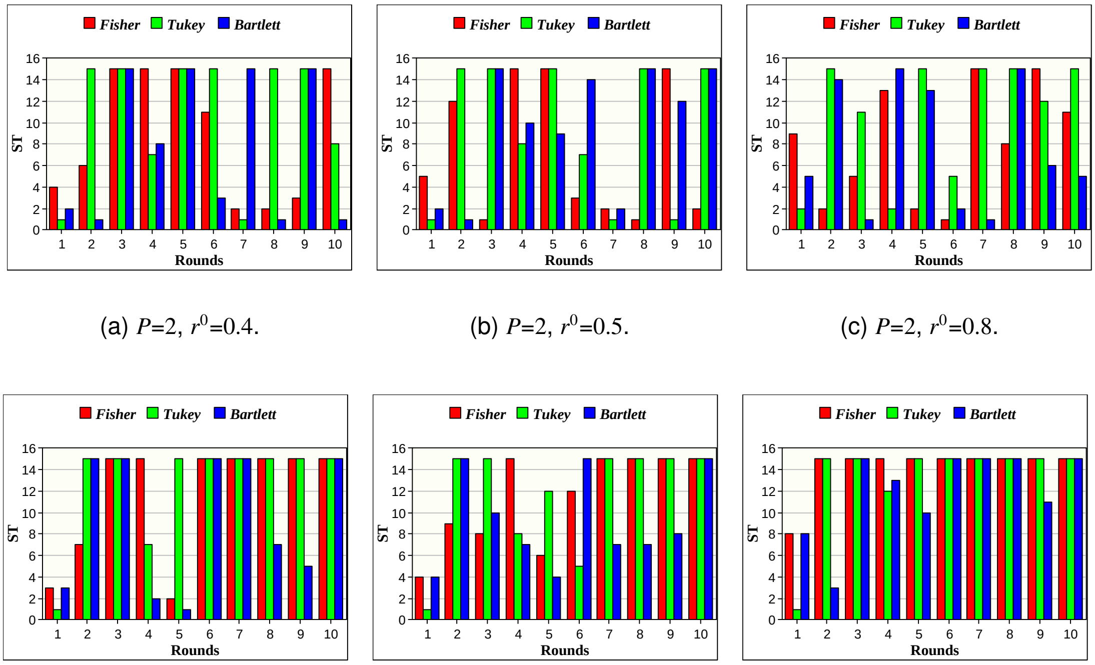
- The instantaneous sampling rate (ST) after each round, this parameter reflects also the percentage of data reduction in PSN,
- The energy consumption of the node,
- The network's lifetime (number of rounds).

2.5.1/ Adaptive Sampling Rate to Data Variance and Application Criticality

In this section we show the results obtained while considering the variance of the collected data and the application criticality. We studied the instantaneous sampling rate adaptation and the overall energy consumption.

2.5.1.1/ Instantaneous Sampling Rate

The main goal of this section is to show, on the one hand, how our approach is able to reduce and to adapt its sampling rate according to the application criticality level, and on the other hand, to compare the results of the three different tests Fisher, Tukey and Bartlett. Figure 2.3 shows the instantaneous sampling rate results for the three tests. In Figures 2.3(a), 2.3(b) and 2.3(c) we fixed each round to two periods ($P = 2$) when we varied the criticality level (r^0) to 0.4, 0.5 and 0.8 respectively, while in Figures 2.3(d), 2.3(e) and 2.3(f) we fixed each round to three periods ($P = 3$) with the same variation values of r^0. Based on these figures, we can see that the three tests successfully adapt the sampling rate of the sensor nodes dynamically after each round according to the application criticality level. These results show how the sampling rate ST varies over time. They confirm also the reduction of the amount of collected data comparing to the nodes operating on $SMAX$ all time. We can also observe that when the risk increases the sampling rate remains usually at its maximum value.



(a) P=2, r^0=0.4.

(b) P=2, r^0=0.5.

(c) P=2, r^0=0.8.

(d) P=3, r^0=0.4.

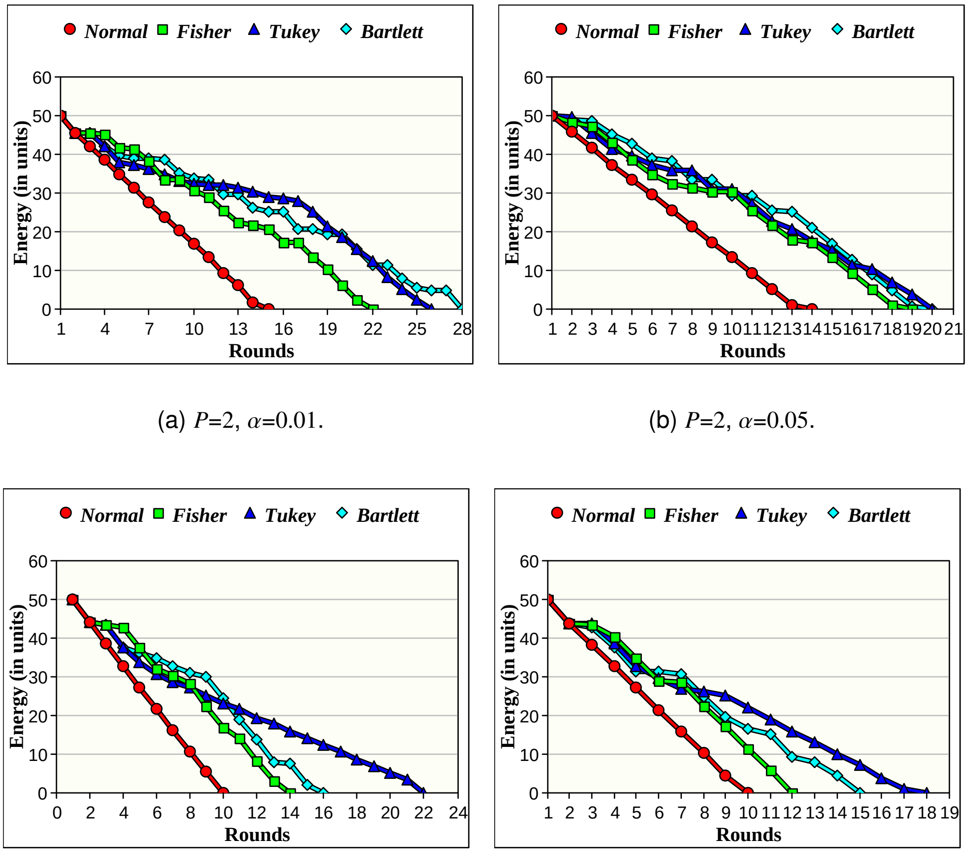
(e) P=3, r^0=0.5.

(f) P=3, r^0=0.8.

Figure 2.3: Variation of sampling rate (ST) over rounds, $SMAX = 15$, $\alpha = 0.05$.

2.5.1.2/ Energy Consumption

The objective of this section is to show how our approach optimizes the energy consumption. Figure 2.4 depicts the results obtained for the three different tests and the non optimized sampling rate which we present it as “Normal” on the figure. In Figures 2.4(a) and 2.4(b), we fixed P to 2 and r^0 to 0.4 and we varied α by giving it the values 0.01 and 0.05. In Figures 2.4(c) and 2.4(d) we fixed P to 3 with same values for r^0 and α. Based on these results, we can notice that our approach in all cases is able to provide a gain of at least 25% of the sensor lifetime comparing to the normal case. This indicates also that our approach can reduce effectively the amount of data collected by the sensors according to the dynamics of the monitored condition.



(a) P=2, α=0.01.

(b) P=2, α=0.05.

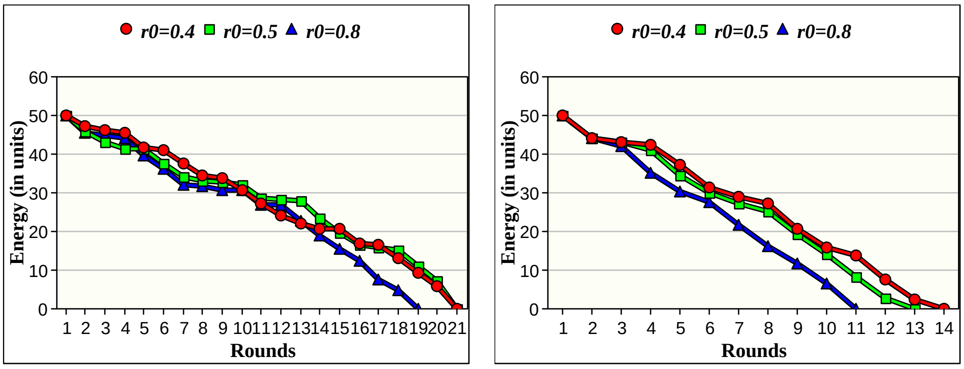
(c) P=3, α=0.01.

(d) P=3, α=0.05.

Figure 2.4: Energy consumption over rounds, $SMAX$=15, r^0=0.4.

From the false-rejection probability (risk α) side, we can see clearly that the energy consumption decreases more when we use $\alpha = 0.05$. Indeed, when the risk α increases the null hypothesis will be able to be more rejected. We can also deduce that when r^0 is fixed then the network's lifetime increases when the number of periods at the round (P) decreases.

Influence of the risk level r^0: In this paragraph, our objective is to show the influence of the application risk level r^0 on the energy consumption. Figure 2.5 illustrates the comparison between the results of the Fisher test while varying r^0 (0.4, 0.5 and 0.8) for the two cases $P = 2$ (Figure 2.5(a)) and $P = 3$ (Figure 2.5(b)). Based on these results, we can see that increasing the sensor lifetime is inversely proportional to the level value risk r^0. This is to confirm that, in the case of critical applications our approach is more careful, and the amount of the collected data becomes more important to ensure that we do not miss any critical reading.



(a) P=2.

(b) P=3.

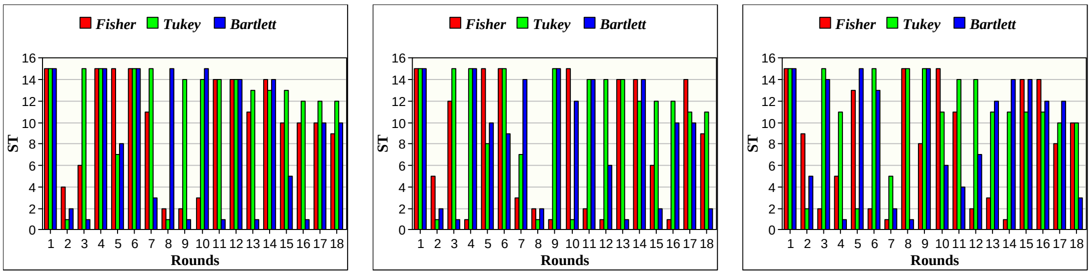
Figure 2.5: Energy consumtpion while varying r^0, $SMAX$=15, α=0.01.

2.5.2/ Adapting Sampling While Considering the Residual Energy Level

In this section we present the results obtained while considering the variance of the collected data, the application criticality and the residual energy. As before, we studied the instantaneous sampling rate adaptation and the overall energy consumption.

2.5.2.1/ Instantaneous Sampling Rate

In this part of experiments, we take into account the residual energy of the nodes to adapt their sampling rates. Figure 2.6 shows the sampling frequency calculated after each round after applying the three tests where we fixed P to 2 and α to 0.05 and varied r^0 to 0.4, 0.5 and 0.8. Based on the obtained results, we can observe that the number of periods where the ST reach S_{max} when applying Bartlett test is less than the other tests which leads to conclude that the Bartlett test will decrease more the consumption of energy more than the other tests.



(a) r^0=0.4.

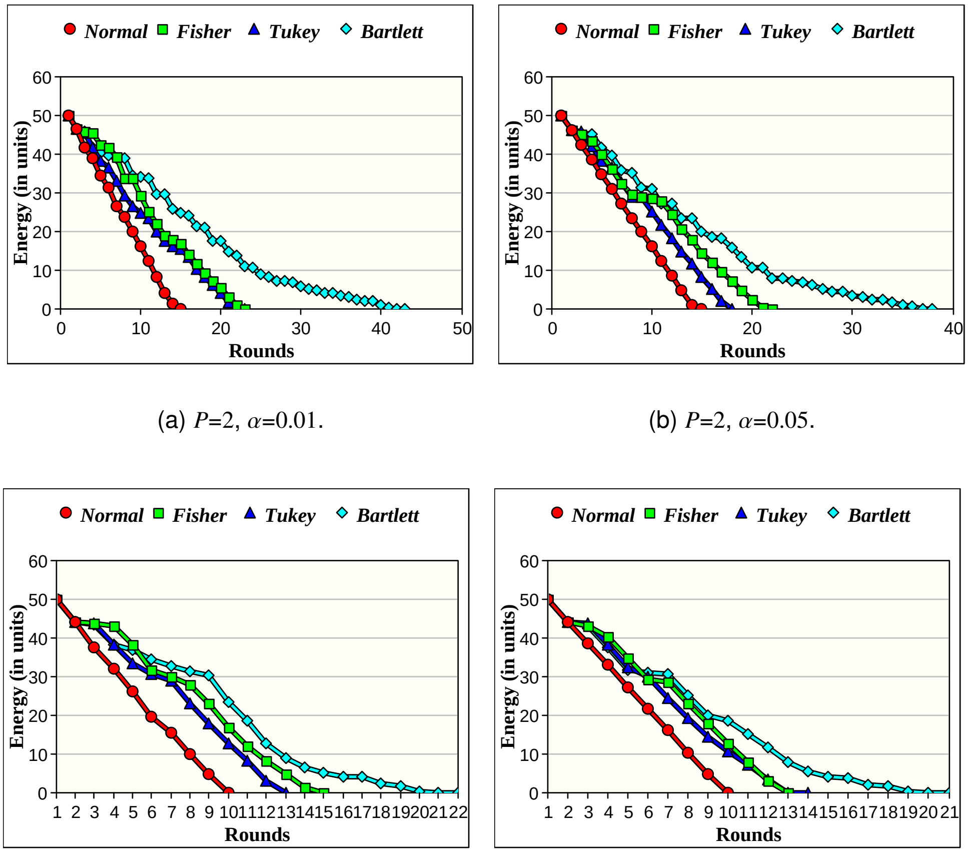
(b) r^0=0.5.

(c) r^0=0.8.

Figure 2.6: Variation of sampling rate (ST) over rounds with residual energy adaptation, P = 2, $SMAX$ = 15 and α = 0.05.

2.5.2.2/ Energy Consumption

Figure 2.7 presents the results of the adaptive model which take the variance of the readings and the residual energy as parameters to calculate the sampling rate of the sensors. We fixed P to 2 in Figures 2.7(a) and 2.7(b) and we varied α to 0.01 and 0.05 respectively, while in Figures 2.7(c) and 2.7(d) we fixed P to 3 with the same values for α. We fixed r^0 to 0.4 in all cases. Comparing the results of this figure to the results of the Figure 2.4 (adaptation without residual energy), we can see that considering the residual energy permits to extend the network lifetime.



(a) P=2, α=0.01.

(b) P=2, α=0.05.

(c) P=3, α=0.01.

(d) P=3, α=0.05.

Figure 2.7: Energy consumption over rounds with residual energy adaptation, $SMAX$=15, r^0=0.4.

2.5.3/ Further Discussions

In this section, we give further consideration to our proposed approach. We compare the obtained results while applying the three different statistical tests. We give some directions to which method to choose under which conditions and circumstances of the application.

From the energy preserving point of view and based on the results presented in Figure 2.4, we can deduce that Tukey and Bartlett tests allow saving more energy than the Fisher test. This is because Tukey and Bartlett tests are more flexible regarding the variance between readings compared to the variance calculated in Fisher. Therefore, if the application permits flexibility regarding the data variance, Bartlett and Tukey tests are more suitable. It is a compromise between energy saving and variance flexibility.

On the other hand, we can also deduce that the Bartlett test extends the network lifetime more than the other tests when P is small, e.g. $P = 2$, (Figures 2.4(a) and 2.4(b)), while Tukey test gives better results when P is large, e.g. $P = 3$, (Figures 2.4(c) and 2.4(d)). Moreover, Bartlett test conserves more energy of the sensors compared to Fisher and Tukey tests for both small and large P, when the adaptive model takes the residual energy to adapt sensor's sampling rate (Figure 2.7).

2.6/ CONCLUSION

In this chapter, we proposed an adaptive sampling approach for energy efficient periodic data collection in sensor networks. First, we presented an existing technique based on one-way ANOVA model applied with Fisher test for adapting sampling rate based on the variance study. Then, we extended such technique to other statistical tests, such as Tukey and Bartlett while taking into account the residual energy level. We showed via simulations that our approach can be effectively used to increase the sensor network lifetime, while still keeping the quality of the collected data high.

3

Energy-Efficient Data Aggregation and Transfer Protocol

Limited battery power and high transmission energy consumption in wireless sensor networks make in-network aggregation and prediction a challenging area for researchers. The most energy consumable operation is transmitting data by a sensor node, comparing it with the energy consumption of in-network computation which is negligible. The energy trade-off between communication and computation provides applications benefit when processing the data at the network side rather than simply transmitting sensor data. In this chapter, we propose energy efficient two phase data aggregation technique for clustering based PSN. The first phase, called aggregation phase, is used to find similarities between data (readings captured during a period p) in order to eliminate redundancy from raw data, thus reducing the amount of data-sets sent to the CH. The second phase, called transmission phase, provides sensors the ability to identify duplicate data sets captured among successive periods, using the sets-similarity joins functions.

3.1/ Introduction

In WSN, sensor node lifetime is highly related to the power consumption of its battery as it is the low source of energy, and it is difficult and cost ineffective to recharge it in most cases. Sensor nodes use their limited energy in computation and transmission processes in a wireless environment, but the power consumption is at the highest level when sending and receiving messages.

Data aggregation and data reduction approaches are proposed to conserve energy in WSN by reducing the amount of data sent from sensor nodes to their appropriate sink. To save overall energy resources, sensing data are aggregated along the route from sensors to sink. In addition, the amount of data generated in large sensor networks is usually redundant which makes the data aggregation methods essential to eliminate redundant transmissions. It is important to highlight that data aggregation, by eliminating redundant data, should not affect the quality of data (e.g. data accuracy and integrity).

In this chapter, we tackle a new area within data aggregation and reduction problems, by focusing on identifying the similarity between sets of data generated by each sensor node. Since sensing data depends on the monitored condition or process, it is likely that sensor nodes generate similar sets of data for many successive periods, especially when the monitored condition is somehow static. Our main objective in this chapter is to reduce the amount of data transmitted from sensor nodes to their sink. Therefore, we study the similarity between sets generated in successive periods. We suggest an energy efficient technique based on the sets-similarity joint functions that conserve data integrity while eliminating inherited redundancy.

Similarity between sets of sensed data can be computed by using similarity functions, which measure the degree of similarity between two sets and return a value between 0 and 1. Higher is the similarity value, more similar are the sets. Therefore, we can treat pairs of sets with high similarity values as redundant sets, thus, the concerned sensor node sends only one set to the appropriate sink instead several similar sets.

The remainder of this chapter is organized as follows. Section 3.2 presents related work on data reduction and aggregation in sensor networks. Section 3.3 presents our technique consists of two phases: aggregation and transmission phases. Experimental results are exposed in section 3.4. Finally, we conclude the chapter in Section 3.5.

3.2/ Data Transmission Reduction: A Background

Reducing data transmission is the main challenge in WSNs since data transmission is the higher energy consumer process for a sensor node [119, 11]. Hence, researchers have focused on the data aggregation as efficient way to reduce the data transmitted in the network, by eliminating redundancy from the raw data and sending only the useful information to the sink. Furthermore, a lot of such data aggregation studies have been made based on clustering schemes, such as DDCD [170] and DUCA [104]. The authors in [143, 73, 103] present a comprehensive overview about different data aggregation techniques based on clustering network architecture proposed in the literature for WSNs.

The authors in [175] propose a Distributed K-mean Clustering (DKC) method for WSN. On the basis of DKC, the authors build a network data aggregation processing mechanism based on adaptive weighted allocation of WSN. DKC algorithm is mainly used to process the testing data of bottom nodes in order to reduce the data redundancy. In [139], the authors propose a data aggregation based clustering scheme for underwater wireless sensor networks (UWSNs) which involves four phases. The goals of these phases are to reduce the energy consumed in the overall network, increasing the throughput, and minimizing data redundancy. The authors in [74] propose a M-EECDA (Multihop Energy Efficient Clustering & Data Aggregation Protocol for Heterogeneous WSN). The protocol combines the idea of multihop communications and clustering for achieving the best performance in terms of network life and energy consumption. M-EECDA introduces a sleep state and three tier architecture for some cluster heads to save energy in the network. The authors in [72] propose two clustering-based protocols for heterogeneous WSNs, which are called single-hop energy-efficient clustering protocol (S-EECP) and multi-hop energy-efficient clustering protocol (M-EECP). In S-EECP, the cluster heads (CHs) are elected by a weighted probability based on the ratio between residual energy of each node and average energy of the network whereas in M-EECP, the elected CHs communicate the data packets to the base station via multi-hop communication approach. In [33], the authors propose an adaptive data aggregation (ADA) scheme for clustered sensor networks. In the latter, a time based technique and spatial aggregation degrees are introduced. They are controlled by the reporting frequency at sensor nodes and by the aggregation ratio at CHs respectively.

Recently, various data aggregation methods [20, 19, 82, 91] have been proposed in the literature used to eliminate the inherent redundancy in raw data collected by periodic sensor networks. The authors in [20, 19] propose a data cleaning pre-processing approach to reduce the packet size transmitted and prepare the data for an efficient data mining technique based on k-means and FP-tree. The idea is to periodically search similarities between all received readings at the aggregator level in order to reduce the size of data by introducing the notion of reading's occurrence. Then, a data mining algorithm (FP-Tree) is applied in order to send only useful information to the sink. In [82], the authors propose a Cluster-based False data Filtering Scheme (CFFS) that can detect and filter out false reports travel in the network before leading to a waste of energy of this network. In [140], the authors use Euclidean distance and cosine distance at the aggregator level to build an efficient underwater network by reducing packet size and by minimizing data redundancy.

In most of existing techniques, we can find data aggregation that focus mainly on the selection of cluster heads and the data transmission to the sink. Furthermore, only the cluster heads process and aggregate data without any processing at the level of the nodes themselves. Recently, the authors in [18] propose the prefix-frequency filtering (PFF) technique which study a new area within filtering aggregation problem in PSN. the objective of PFF is to identify similarity between data at both sensors and cluster heads (CHs). At the first level of aggregation, e.g. at sensor level, each sensor node searches the similarity between sensed data before sending them to its appropriate CH. When the CH receives data sets from all its nodes, it searches the similarity between data generated by neighboring sensor nodes at the second level of aggregation, by using similarity sets functions. Then, in order to avoid the comparisons between all the received sets, they provided several optimizations aiming to enhancing the data latency of the PFF [21, 22]. Recently, PFF technique can be considered as one of the efficient data aggregation technique in PSN because it conserves energy for both sensors and CHs. However, PFF has two disadvantages: first, it allows each sensor node to eliminate redundancy from raw data in each period and not in successive periods; second, it has a heavy computational load when searching similar data sets generated by neighboring sensor nodes at the aggregator level.

3.3/ Data Aggregation and Transfer in PSN

In this section, we describe our technique at the sensor node consists of two phases: aggregation and transmission. The objective of these phases is to eliminate the redundancy existing among the collected data thus, reduce the amount of data sent from each sensor to its CH.

3.3.1/ First Phase: Aggregation Phase

In PSN, it is very likely to happen that sensor nodes collect the same or very similar consecutive data. For instance, we take two examples of networks, one deployed in the Intel Laboratory [88] which collects temperature readings (Figure 3.1) and the second is deployed in the Indian Ocean [115] which collect salinity readings (Figure 3.2). For a period of one day, readings collected by sensors S_{10} and S_{46}, in Figure 3.1, span over a range of $[15.95, 21.45]$ and $[14.01, 22.16]$ respectively. On the other hand, $S_{1901149}$ and $S_{1901332}$ collect readings over a range of $[34.32, 34.78]$ and $[33.82, 35.27]$ respectively for the same period. Small ranges of measures shown in Figure 3.1 and Figure 3.2 indicate that readings collected by each sensor are very redundant in this period. Therefore, if sensors send all the collected measures to their appropriate CHs, their energy will be wasted and thus the whole network energy will be depleted quickly. Hence, data aggregation becomes an important requirement in WSNs in order to minimize redundant data collected by the sensor nodes.

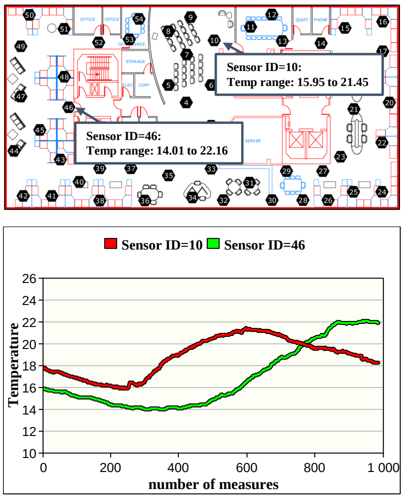


Figure 3.1: Sensors in the Intel Laboratory.

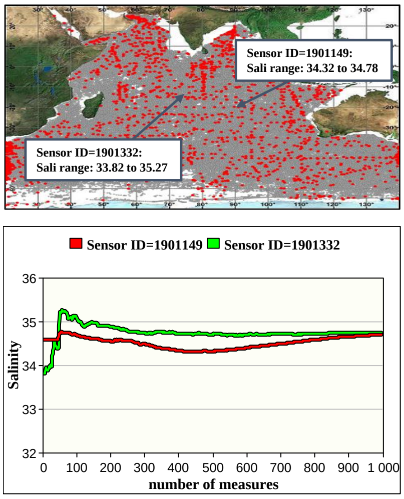


Figure 3.2: Sensors in the Indian Ocean (Argo project).

3.3.1.1/ Definitions and Notations

We denote by $S=\{1, 2, \ldots, N\}$ the set of sensor nodes, where N is the number of nodes. Then, each period p in PSN is divided into $\mathcal{T}$ equal time slots where, at each slot, a sensor S_i captures a new reading r_i, then, it forms a vector of readings during the period p as follows: $R_i=[r_{i_1}, r_{i_2}, \ldots, r_{i_\mathcal{T}}]$. Sensor S_i in Figure 3.3 takes five measures (e.g. $\mathcal{T}=5$), at each period p_q ($q \in [1,3]$) and sends its vector of collected data $R_i=[r_{i_1}, r_{i_2}, r_{i_3}, r_{i_4}, r_{i_5}]$ to the CH at the end of the period.

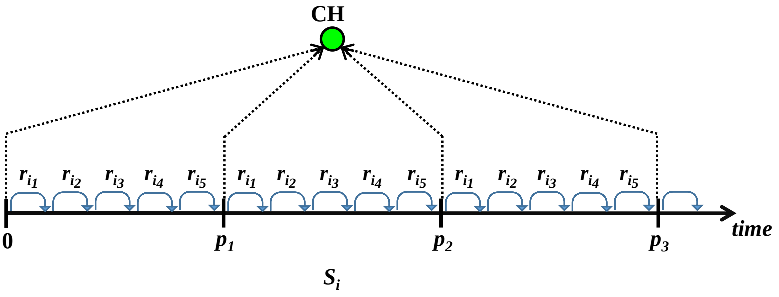

CH
r_{i_1} r_{i_2} r_{i_3} r_{i_4} r_{i_5}
r_{i_1} r_{i_2} r_{i_3} r_{i_4} r_{i_5}
r_{i_1} r_{i_2} r_{i_3} r_{i_4} r_{i_5}
0
p_1
p_2
p_3
time
S_i

Figure 3.3: Data collection in PSN.

As mentioned above, a data vector R_i formed by the sensor S_i may contain redundant (or very similar) readings, especially when the monitored condition varies slowly or when the slots are short. In order to eliminate similar values from the vector R_i, the authors in [18] proposed the $link$ function to search for readings similarity in the vector. Then, they assigned to each reading its number of occurrences in R_i. Similar to $link$ function, we define $Similar$ function as follows:

Definition 3.1 *$Similar$ function. We define the $Similar$ function between two readings captured by the same sensor node S_i at a period p as:*

$$Similar(r_{i_j}, r_{i_k}) = \left\{ \begin{array}{ll} 1 & \text{if} \quad \left\| r_{i_j} - r_{i_k} \right\| \leq \delta, \\ 0 & \text{otherwise.} \end{array} \right\}.$$

where r_{i_j} and $r_{i_k} \in R_i$ and δ is a threshold determined by the application. Furthermore, two readings are considered similar if and only if their $Similar$ function is equal to 1.

In order to save the integrity of the information, we define the weight of a reading as follows:

Definition 3.2 *Reading's weight, $wgt(r_i)$. The weight of a reading r_i is defined as the number of similar readings (according to the $Similar$ function) to r_i in the same vector R_i.*

3.3.1.2/ Aggregation Phase Algorithm

Using the notations defined above, we present the aggregation algorithm which is running by the sensors themselves at each period (see Algorithm 2). In the first slot at the period p, the sensor node S_i takes the first reading, initializes its weight to 1 and adds it to the final set which will be sent to the CH. Then, for each new captured reading, S_i searches for similarities of the new taken reading. If a similar reading is found, the new one is deleted and the corresponding weight is incremented by 1, else the sensor adds the new reading to the set and initializes its weight to 1.

After applying Algorithm 2, S_i will possess a set of readings associated to their corresponding weights as follows: R'_i=$\{(r'_{i_1}, wgt(r'_{i_1})), (r'_{i_2}, wgt(r'_{i_2})), \ldots, (r'_{i_k}, wgt(r'_{i_k}))\}$, where $k \leq \mathcal{T}$.

Based on the set R'_i, we provide the following definitions:

Definition 3.3 *Cardinality of the set R'_i, $|R'_i|$. The cardinality of the set R'_i is equal to the number of elements in R'_i, i.e. $|R'_i| = k$.*

Definition 3.4 *Weighted Cardinality of the set R'_i, $wgt_c(R'_i)$. The weighted cardinality of the set R'_i is equal to the sum of all readings' weights in R'_i as follows: $wgt_c(R'_i) = \sum_{j=1}^{|R'_i|} wgt(r_{i_j})$=$\mathcal{T}$, where $r_{i_j} \in R'_i$.*

At the end of each period p, each sensor node S_i will have a set R'_i with no redundant readings. Then, it runs the second phase in our technique: Transmission Phase.

3.3.2/ Second Phase: Transmission Phase

At this step, each sensor node S_i after forming its set of readings at the end of the first phase, decides whether or not to send the set of readings to the CH based on previous readings. During this phase each sensor identifies the similarity between sets collected

Algorithm 2: Aggregation Phase Algorithm

```
Data: new reading $r_i$
Result: set of readings with their weights: $R'_i$
$R'_i \leftarrow \emptyset$;
if $j = 1$ // first reading at period $p$ then
    $wgt(r_{i_j}) \leftarrow 1$;
    $R'_i \leftarrow R'_i \cup \{(r_{i_j}, wgt(r_{i_j}))\}$;
end
else
    $found \leftarrow false$;
    while $((r_{i_k}, wgt(r_{i_k})) \in R'_i)$ && $(!found)$ do
        if $Similar(r_{i_j}, r_{i_k}) = 1$ then
            $wgt(r_{i_k}) \leftarrow wgt(r_{i_k}) + 1$;
            disregard $r_{i_j}$;
            $found \leftarrow true$ ;
        end
    end
    if $(!found)$ then
        $wgt(r_{i_j}) \leftarrow 1$;
        $R'_i \leftarrow R'_i \cup \{(r_{i_j}, wgt(r_{i_j}))\}$;
    end
end
```

during successive periods to adapt the transmission of sets to the CH. In case of successive periods are similar, the sensor doesn't need to send sets in all the periods. Instead, it sends a notification indicating that the sets are similar in order to conserve its energy therefore, the CH must use the last set sent (e.g., is not notification) by the sensor to be the current set.

In PSN, there is a couple of important design considerations associated with the periodic data model. Sometimes, the dynamics of the monitored condition or process can slow down or speed up [76]; in case of slow down, the sensor will forward more redundant data to the CH, especially when period p is short. If the sensor can adapt its data transmission to the changing dynamics of the condition or process, then data sets transmitted to the CH can be minimized, and power efficiency of the overall network system can be improved. Another critical design issue is the relation among multiple sensor nodes. Since the collection of data is periodic, collision occurs between packets exchanged between nodes in the network especially when the network is loaded. It is essential for sensor nodes to be able to detect collisions and to introduce a phase shift between two transmission sequences in order to avoid further collisions [76]. Therefore, the main goal of the second phase in our proposal is to let the sensor detects similarity between data sets captured in successive periods using sets similarity functions. Thus, the number of sets sent from sensor nodes to their appropriate CHs will be reduced as well as the collision between packets and the bandwidth on the network.

3.3.2.1/ Similarity Functions

In the literature, similarity functions were used in various domains and applications in order to identify near duplicate objects (data), such as web search engines [54], Web mining applications [29], detecting plagiarism [56], collaborative filtering in data mining [26], etc. Thus, similarity functions can also constitute a suitable solution for PSNs where data collected by the sensors are in form of sets and they are usually redundant. Hence, the authors in [22] [21] and [18] are the first that apply the similarity functions in WSN in order to find neighbor nodes that generate similar data. They use similarity functions at the level of aggregators. The novelty of our work is that we use the similarity function at the level of sensor nodes instead of aggregators.

Similarity functions, $Sim(R_i, R_j)$, calculate similarity between two given sets. Functions return a value between 0 and 1; if the value was 0 then sets are entirely different, while if the value is equal 1, then it means that sets are identical. All other value between 0 and 1 describe the level of similarity between sets. Two sets are similar if their calculated Sim is greater than a given threshold t. Several functions have been proposed in the literature in order to measure the similarity between two data sets R_i and R_j such as:

$$\begin{array}{ll}
\text{Overlap similarity:} & O(R_i, R_j) = |R_i \cap R_j| \\
\text{Jaccard similarity:} & J(R_i, R_j) = \frac{|R_i \cap R_j|}{|R_i \cup R_j|} \\
\text{Cosine similarity:} & C(R_i, R_j) = \frac{|R_i \cap R_j|}{\sqrt{|R_i| \times |R_j|}} \\
\text{Dice similarity:} & D(R_i, R_j) = \frac{2 \times |R_i \cap R_j|}{|R_i| + |R_j|}
\end{array}$$

However, in this work, we are interested in Jaccard similarity function for two reasons: first, it is one of the most widely used functions as it can support any other similarity functions [12]; second, to compare to other methods using the same function. The Jaccard similarity of sets R_i and R_j is $J(R_i, R_j)$, which is the ratio of the size of the intersection between R_i and R_j to the size of their union. Figure 3.4 shows an example of calculation of J between two sets R_i and R_j. There are three elements in their intersection and a total of eight elements that appear in R_i,R_j or both. Thus, $J(R_i, R_j) = 3/8$.

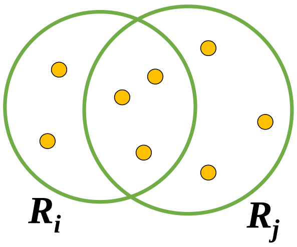


Figure 3.4: Two sets with Jaccard similarity $3/8$.

Adapting Jaccard similarity to readings weights, two given sets R'_i and R'_j are considered similar if and only if:

$$J(R'_i, R'_j) \geq t$$

where t is a threshold given by the application itself.

For two similar readings $r'_i \in R'_i$ and $r'_j \in R'_j$ such that $Simlar(r'_i, r'_j) = 1$, we denote $wgt_{min}(r'_i, r'_j) = min(wgt(r'_i), wgt(r'_j))$ the minimum value of the weight of these readings.

Then, inspired from [18], we define "$\cap_s$" as a new function for overlapping as follows:

Definition 3.5 *Consider two sets of readings R'_i and R'_j, then we define: $R'_i \cap_s R'_j = \{(r'_i, r'_j) \in R'_i \times R'_j$ with weight $wgt_{min}(r'_i, r'_j)$, such that $Similar(r'_i, r'_j) = 1\}$ and we consider it remains:*

$$\left\{ \begin{array}{lll} \text{in } R'_i : & R'_i - \{(r'_i, wgt(r'_i) - wgt_{min})\} & if\, wgt(r'_i) - wgt_{min} > 0, \\ or & & \\ \text{in } R'_j : & R'_j - \{(r'_j, wgt(r'_j) - wgt_{min})\} & if\, wgt(r'_j) - wgt_{min} > 0. \end{array} \right\}.$$

Now, the Jaccard similarity function between two sets R'_i and R'_j can be defined as follows [21]:

$$J(R'_i, R'_j) \geq t \Leftrightarrow |R'_i \cap_s R'_j| \geq \alpha = \frac{2 \times t \times \mathcal{T}}{1 + t} \tag{3.1}$$

Algorithm 3 searches the similarity between two sets given by its arguments, it returns a boolean value which indicates if the sets are similar or not.

Algorithm 3: Jaccard Similarity Sets Algorithm

Data: two sets of readings R'_i and R'_j, t, $\mathcal{T}$
Result: boolean value that indicates if the sets are similar

```
$O_s \leftarrow 0$;
Consider $|R'_i| < |R'_j|$;
for $k \leftarrow 1$ to $|R'_i|$ do
    search similar of $R'_i[k]$ in $R'_j$;
    find $R'_j[l]$ / $Similar(R'_i[k], R'_j[l]) = 1$;
    if $R'_j[l]$ is exist then
        $O_s \leftarrow O_s + wgt_{min}(R'_i[k], R'_j[l])$;
    end
end
if $O_s \geq \frac{2 \times t \times \mathcal{T}}{1+t}$ then
    return $true$;
end
else
    return $false$;
end
```

3.3.2.2/ Optimization of Jaccard Similarity Computation

In WSNs, the set of collected readings can contain a large number of elements making the Jaccard similarity function very expensive in terms of calculation. Thus, data sets arrived to the sink node will be delayed. Hence, in order to reduce the overhead of the Jaccard function, the authors in [21] proposed a filtering constraint during the verification of the similarity between two data sets. Thus, in this chapter, we use such filtering in order to accelerate the computation of the Jaccard similarity function. The proposed filtering divides two compared sets in order to find a reading where in its position a similarity upper bound is estimated and checked against the similarity threshold. As soon as the check is failed we can stop the overlap computing early. The proposed filtering is formalized by the following lemma [21]:

Lemma 3.1 *Assume that* $|R'_i| < |R'_j|$ *and all readings in* R'_i *are ordered according to the global ordering* O*.* R'_i *and* R'_j *are similar* $\Rightarrow$ *for any* $r' \in R'_i$ *dividing* R'_i *into* $h\text{-}R'_i$ *and* $l\text{-}R'_i$ *we have:* $|h\text{-}R'_i \cap_s R'_j| \geq (2 \times t \times \mathcal{T})/(1+t) - \sum_{k=1}^{|l\text{-}R'_i|}(wgt(r'_k \in l\text{-}R'_i))$*.*

Proof. Please refer to the lemma 2 in [21] to see the proof. □

3.3.2.3/ Data Sent during Transmission Phase

After forming the final data set at the end of the first phase, each sensor node executes the transmission phase in order to decide either to transfer or not the data set to the CH. In the transmission phase, each sensor node sends at each period one of the following packets: Set_Packet or $Similarity\_Notification$. The first one contains the current data set formed at the current period while the second is an empty packet. However, a sensor node computes the similarity between the current data set and the last data set sent (e.g. in the last Set_Packet packet which is saved in its memory) to the CH, by using Jaccard function mentioned above. If the similarity is greater than the Jaccard threshold, then sensor node sends a $Similarity\_Notification$ packet to the CH in order to avoid sending redundant data sets in several periods, (in order to decrease the power consumption). In the other case, (i.e. the similarity is less than the Jaccard threshold), the sensor node replaces the saved set in its memory by the current set, then sends it in a Set_Packet packet to the CH.

Next, we provide an example that illustrates the transmission phase. In this example, we present data transmission by a sensor ‘S_1’ for five successive periods (p_1 to p_5) after aggregation phase at each once (Figure 3.5).

In the first period p_1, the sensor S_1 sends the set R'_1 to the CH after saving a copy in its memory. At the second period p_2, S_1 checks similarity between the new set R'_2 and the set saved in its memory (e.g., R'_1). It detects that both sets are similar, then it deletes the new set R'_2 and sends a notification to the CH. At period p_3, S_1 checks similarity between R'_3 and R'_1 which is saved in its memory; it detects that R'_1 and R_3 are similar, therefore, it keeps R'_1 saved and deletes R'_3 while sending a notification to CH. The same process is applied at period p_4, but in this case R'_1 and R'_4 are not similar, thus, S_1 replaces the set saved in its memory (e.g., R'_1) by the new one R'_4 and sends R'_4 in a Set_Packet to the CH. The same process is applied on data transmission between the CH/sink and the user in order to reduce unnecessary transmissions, while as mentioned before, the sink sends the set of readings received in the last Set_Packet at the end of each period even if notification packets are received.

3.3.2.4/ Algorithms Used for Transmission Phase

Based on the example described at previous section 3.3.2.3, we present in this section the set of algorithms used by the network devices during the transmission phase. Algorithm

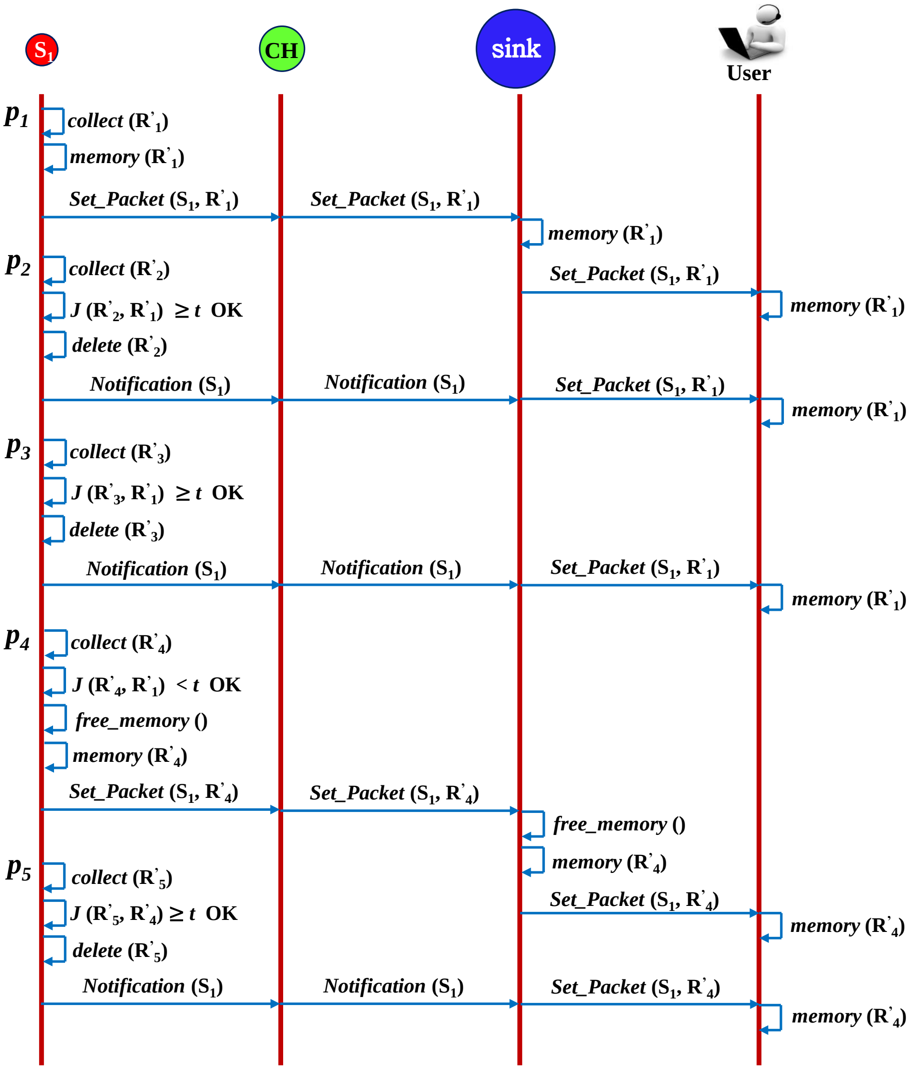


Figure 3.5: Illustrative example for transmission phase.

4, 5 and 6 below are used to adapt the transmission of data between network devices.

Algorithm 4 describes the decision phase at the sensor side. In the first period, the sensor saves and sends the set of readings to the CH (lines 2–3), then at the end of each period, the sensor sends a set of readings if and only if the set is not similar to the saved set (lines 11–13), else it deletes the set and sends a notification to its proper CH (line 7–8).

Algorithm 5 describes how the CH receives and sends the data sets sent by its proper sensor nodes. However, since our technique can be applied in a distributed manner at each sensor separately, the main responsibility of the CH is to manage the nodes in its cluster. Furthermore, the CH has been considered in our technique as an intermediate node that manage data transmission between the sensor nodes and the sink. Therefore, when the CH receives a packet, Set_Packet or $Similarity\_Notification$, from a sensor node it forwards it directly to the sink.

In order to preserve the information integrity, the sink saves the received set sent

Algorithm 4: Transmission Phase Algorithm at Sensor

Data: sensor id, R'_i: current set of readings at period p, R'_j: saved set of readings, t, $\mathcal{T}$
Result: saved set R'_j

```
if $p$ is the first period then
    memory($R'_i$) // save current set in memory;
    $Set\_Packet(id, R'_i)$ // send new set to CH;
end
else
    if $SimilaritySets(R'_i, R'_j, t, \mathcal{T})$ then
        delete $R'_i$;
        $Similarity\_Notification(id)$ // send a notification to CH;
    end
    else
        free_memory() // delete saved set from memory;
        memory($R'_i$);
        $Set\_Packet(id, R'_i)$;
    end
end
```

Algorithm 5: Transmission Phase Algorithm at CH

Data: packet sent from sensor id at period p
Result: void

```
if $Set\_Packet(id, R'_{id})$ at $p$ then
    $Set\_Packet(id, R'_{id})$ // send the set received from sensor $id$ to sink;
end
else
    $Similarity\_Notification(id)$ // send an empty packet corresponding to sensor $id$ to sink;
end
```

in the last Set_Packet of each sensor in its memory. In case the sink recognizes that $Similarity\_Notification$ packet is received from the sensor at the current period, then, it uses the saved set of readings of the concerned sensor, as the current set. While in the other case, (e.g., sink recognizes that Set_Packet is received from the concerned sensor) the sink replaces the saved set of the concerned sensor by the received one. At the end of each period, the sink sends the last sets for all sensors saved in its memory to the final user in order to achieve data accuracy.

Algorithm 6 describes data transmission at the sink side. The sink has a map I_{id} to save last sets in last Set_Packet received from all sensor nodes. When the sink receives a new data set from a sensor, it replaces the saved set by the new set. Then, it sends all the data sets saved in I_{id} at each period to the user in order to guarantee real time information.

Algorithm 6: Transmission Phase Algorithm at Sink

Data: packet sent from sensor id at period p, I_{id}: Map from sensors id to corresponding saved sets R'_{id}

Result: void

```
for each $Set\_Packet(id, R'_{id})$ at $p$ do
    replace($I_{id}$, $id$, $R'_{id}$) // replace the saved set by the new set;
end
for each set $R'_{id} \in I_{id}$ do
    $Set\_Packet(id, R'_{id})$ // send last saved set for each sensor $id$ to user;
end
```

3.3.3/ Combining of First and Second Phases at the Sensor Level

In this section, we integrate phases (aggregation and transmission) proposed in our technique at the sensor level. Algorithm 7 provides our technique that allows to each sensor, at each period, to eliminate similar captured readings at the aggregation phase, and then to reduce the number of sets sent to its proper CH at the transmission phase.

Algorithm 7: Our technique

Data: t, $\mathcal{T}$, saved set of measures R'_j

Result: void

```
$R'_i \leftarrow \emptyset$;
for $k \leftarrow 1$ to $\mathcal{T}$ do
    Get a reading $r_i$;
    $R'_i = Aggregation(r_i)$;
end
$Tranmsission\_Phase\_at\_Sensor(Sensor$ id$, R'_i, R'_j, t, \mathcal{T})$;
```

3.4/ Experimental Results

To validate our proposed technique, we developed a Java based simulator that is run on the data collected from 54 sensors deployed in the Intel Berkeley Research Lab [88]. In this dataset, Mica2Dot sensors with weather boards collect humidity, temperature, light and voltage values. The data were collected using TinyDB in-network query processing system built on the TinyOS platform. The sensor IDs range from 1-54. Figure 3.6 shows a map of the placement of sensors in the lab. We used a file that includes a log of about 2.3 million readings collected from these sensors Data from some sensor nodes may be missing or truncated (yellow sign in Figure 3.6). In the remainder and for the sake of simplicity we are only interested in the temperature[1] field. We assume that the network contains one CH located at the center of the lab where the sensors send periodically their data to this CH. Our goal is to demonstrate that our technique can successfully achieve desirable results in decreasing the power consumption of a PSN. Each node reads periodically real readings saved in that file while applying the aggregation phase. At the end

[1] the others are done by the same manner.

of the first phase, each node decides to send or not the set of collected readings/weights to the CH using the transmission phase. In our simulations we tackled the approach performance using the following parameters: the threshold of the Jaccard similarity function t, the threshold of the $Similar$ function δ, and the number of readings $\mathcal{T}$ taken by each sensor during a period. We evaluated our approach while taking into account the following metrics: the data aggregation ratio at the first phase, the percentage of data sets sent to the CH at the second phase, data accuracy, and energy consumption. Furthermore, in our experiments we compare our proposed technique to a classical clustering approach without aggregation at the node level and then to the most recent published version of the PFF technique [22].

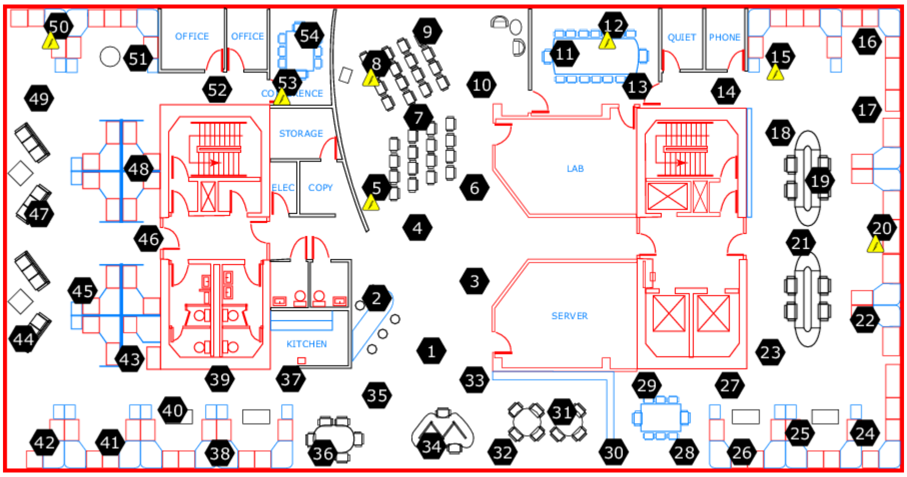


Figure 3.6: Intel Berkeley lab sensor network (image courtesy).

3.4.1/ Data Aggregation Ratio after the Aggregation Phase

Due to the $Similar$ function, each sensor node has the ability to reduce the amount of data collected at each period by eliminating redundant values. Therefore, the result in this phase depends on the chosen threshold δ, the number of the collected measures in period $\mathcal{T}$ and changes in the monitored condition. In these simulations, we vary δ between 0.07 and 0.2 [2], and $\mathcal{T}$ between 20 and 100. Figure 3.7 shows the percentage of the remained readings, or data aggregation ratio, at each period without (classical clustering) and with aggregation phase at each sensor. The results show that, at each period, a maximum of 22% of the data remains after the aggregation phase is applied while the percentage is equal to 100% without applying the aggregation phase. Therefore, first phase can successfully eliminate redundant readings collected by each sensor at each period. We can also observe that, at the aggregation phase, data redundancy increases when $\mathcal{T}$ or δ increases. This is because, $Similar$ function will find more similar readings to be eliminated in each period.

[2] It is chosen in function of the collected readings

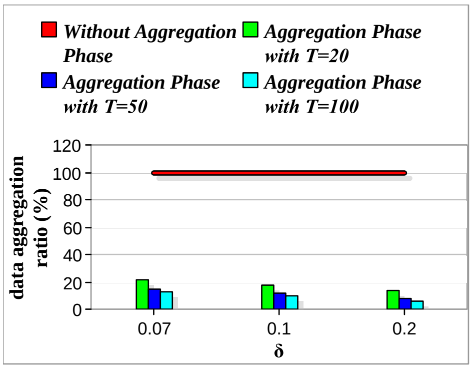
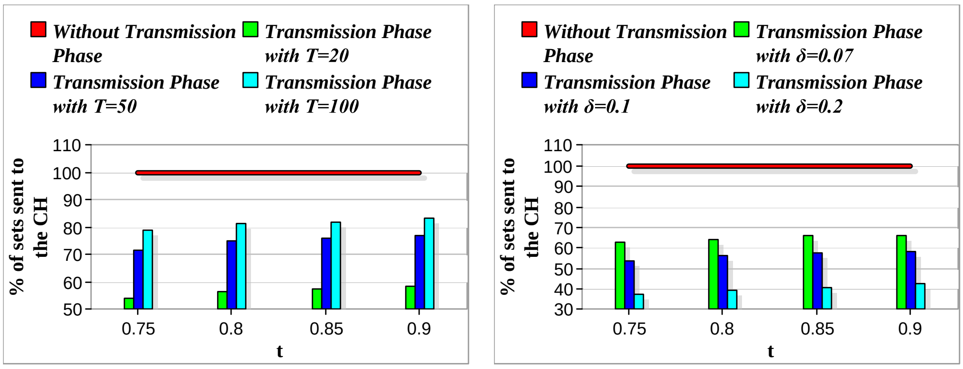


Figure 3.7: Data aggregation ratio after the first phase.

3.4.2/ Percentage of Sets Sent to the CH after the Transmission Phase

In the transmission phase, each sensor reduces the number of sets sent to its proper CH based on the similarity between collected sets of readings among successive periods. In this section, we compare the percentage of sets sent by a sensor with and without applying the transmission phase. We fix in Figure 3.8(a) the threshold δ and vary $\mathcal{T}$ while in Figure 3.8(b) we fix $\mathcal{T}$ and vary δ. The obtained results show that each sensor reduces, when varying the threshold t, 17% to 62% of sets sent to the CH.



(a) $\delta = 0.1$.

(b) $\mathcal{T} = 20$.

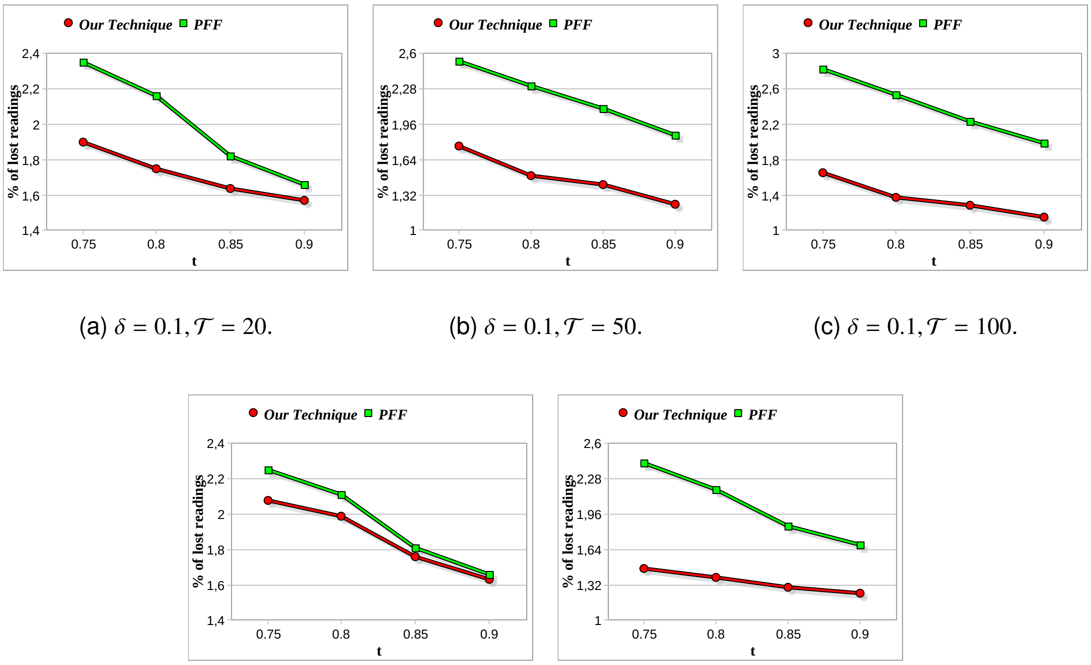
Figure 3.8: % of sets sent to the CH.

Based on results in Figure 3.8, several observations can be made in the transmission phase at the sensor:

- the sensor sends more sets when t increases.
- the sensor eliminates more sets when δ increases.
- transmission phase is more effective when using short periods (e.g., $\mathcal{T}$ decreases).

3.4.3/ Data Accuracy

Data accuracy is an important factor to be considered in WSN, because it affects the decision making by the end user. It represents the measures loss rate. It is an evaluation of measures taken by the sensors nodes and did not arrive neither their similar values to the sink. It is defined also as the aggregation error. In this section we compare the results of our approach to those of PFF technique [22]. Figure 3.9 shows the results of data accuracy for both techniques. In Figure 3.9(a), 3.9(b) and 3.9(c), we fixed the threshold δ and we varied $\mathcal{T}$ while in Figures 3.9(d), 3.9(a) and 3.9(e) we show the comparisons between the two techniques when $\mathcal{T}$ is fixed and δ is varied.



(a) $\delta = 0.1, \mathcal{T} = 20$.

(b) $\delta = 0.1, \mathcal{T} = 50$.

(c) $\delta = 0.1, \mathcal{T} = 100$.

(d) $\mathcal{T} = 20, \delta = 0.07$.

(e) $\mathcal{T} = 20, \delta = 0.2$.

Figure 3.9: Data accuracy.

The obtained results show that the two techniques provide good performance regarding the data accuracy for different values of parameters. We can observe that our technique outperforms PFF in all cases. We can also notice that in our technique, the percentage of readings not received by the sink in worst case (i.e. $t = 0.75$, $\delta = 0.07$ and $\mathcal{T}= 20$ in Figure 3.9(d)) does not exceed the 2.1%. This amount is negligible compared to the amount sent to the end user (the amount of data removed does not affect the user decision making based on the received data). Therefore, we can consider that our technique decreases the amount of redundant data forwarded to the CH and performs an overall lossless process in terms of information and integrity by conserving the weight of each reading. We can also observe that, the percentage of lost readings in our technique decreases when $\mathcal{T}$ increases (Figure 3.9(a), 3.9(b) and 3.9(c)), due to the similarity technique which is more effective and more sets are eliminated when periods are short.

3.4.4/ Energy Consumption Study

Reducing the amount of data transmitted will eventually lead to reduce energy consumption of the sensor and extend its lifetime. Our technique reduces the overhead by aggregating readings at the first phase and adapting data transmission at the second phase, while preserving the information integrity. To evaluate the energy consumption we use the same radio model as discussed in [88]. In this model, a radio dissipates $E_{elec} = 50nJ/bit$ to run the transmitter or receiver circuitry and $\beta_{amp} = 100pJ/bit/m^2$ for the transmitter amplifier. Radios have power control and can expend the minimum required energy to reach the intended recipients as well as they can be turned off to avoid receiving unintended transmissions. Equations used to calculate transmission costs and receiving costs for a k-bit messages and a distance d are respectively shown in 3.2 and 3.3:

$$E_{TX}(k, d) = E_{elec} \times k + \beta_{amp} \times k \times d^2 \tag{3.2}$$

$$E_{RX}(k, d) = E_{elec} \times k \tag{3.3}$$

Receiving is also a high cost operation, therefore, the number of receptions and transmissions should be decreased. With these radio parameters, when d^2 is $500m^2$, the energy spent in the amplifier part is equal to the energy spent in the electronics part, and therefore, the cost to transmit a packet will be twice the cost to receive.

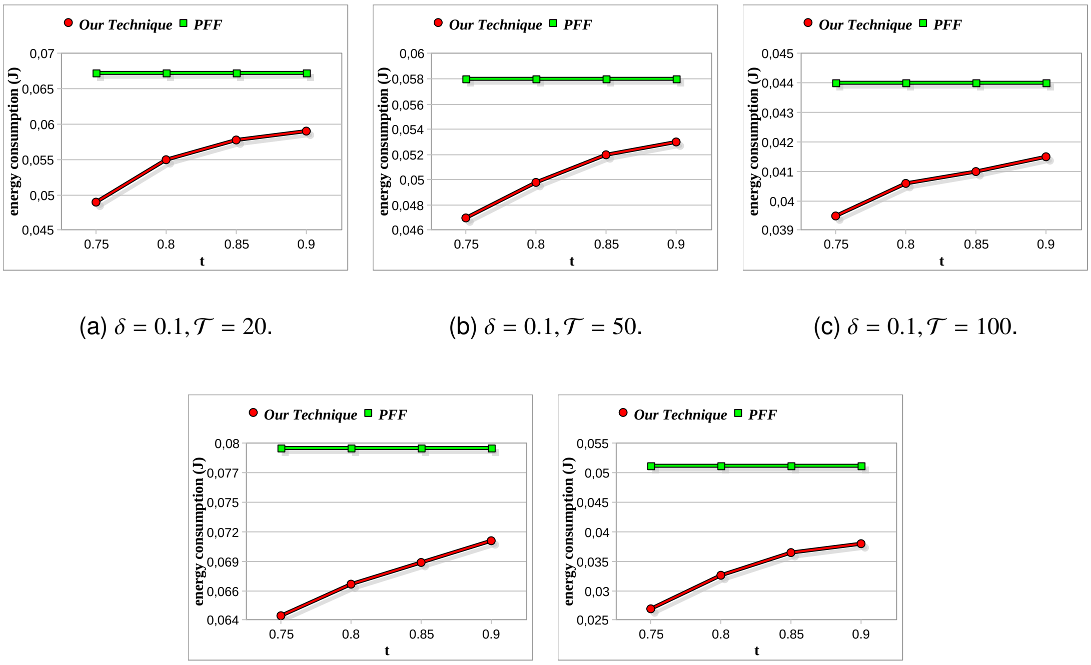
Since our technique also optimizes energy consumption at CH and sink due to minimizing the reception (Equation 3.3) and transmission of sets (Equation 3.2), we only present in this chapter the optimization of energy consumption at sensors level. The goal of the energy study is to show that our technique succeed to conserve energy of sensors and to extend their lifetime more than the PFF technique.

At the end of each period, each sensor forms a set that contains $|R'_i|$ readings with the weight $wgt(r'_i)$ of each one. The size of the set sent by a sensor is equal to the number of weights sent in addition to the number of readings sent. We consider that each reading is equal to 64 bits. Figure **??** show the energy consumption comparison between our technique and the PFF one while varying δ and $\mathcal{T}$.

The obtained results show that our technique outperforms PFF for all values of thresholds and it reduces from 6% to 47% of the energy consumption at each sensor node. This is due to the local transmission phase proposed by our technique, while the PFF sends all the formed sets to the CH.

From these results we can deduce that:

- our technique reduces more energy consumption when t decreases,
- our technique conserves more energy when $\mathcal{T}$ decreases (Figure 3.10(a), 3.10(b) and 3.10(c)) or when δ increases (Figure 3.10(d), 3.10(a) and 3.10(e)).



(a) $\delta = 0.1, \mathcal{T} = 20$.

(b) $\delta = 0.1, \mathcal{T} = 50$.

(c) $\delta = 0.1, \mathcal{T} = 100$.

(d) $\mathcal{T} = 20, \delta = 0.07$.

(e) $\mathcal{T} = 20, \delta = 0.2$.

Figure 3.10: Energy consumption in each sensor.

3.5/ Conclusion

In this chapter we proposed a two phase data aggregation technique based on clustering approach for energy efficiency in PSN: aggregation phase and transmission phase. In the first one each sensor aggregates captured readings based on a $Similar$ function while in the second phase our objective is to reduce the number of data sets sent from sensors to their CHs by searching similarity between captured readings among successive periods using sets similarity functions. The effectiveness of our technique in terms of data reduction and energy consumption is shown through simulations on real data readings. Furthermore, it was shown that our technique outperforms the existing PFF technique dedicated to data aggregation in PSN.

4

In-network Data Aggregation Technique

In WSNs, data transmission is an expensive issue in terms of energy. Therefore, data aggregation becomes an essential technique to achieve energy efficient data transmission for such networks. Energy efficiency, data latency and data accuracy are the major key elements evaluating the performance of a data aggregation technique. The trade-offs among them largely depends on the specific application. In this chapter, we propose a complete data aggregation framework for cluster-based periodic sensor networks. We propose energy-efficient technique which are applied at each cluster separately and achieve aggregation at both sensor nodes and CH levels. Further to a local aggregation at sensor nodes level, we allow each CH to eliminate redundant data sets generated by neighboring nodes at each period by applying distance functions, such as Euclidean and Cosine.

4.1/ Introduction

Wireless sensor networks (WSNs) are almost everywhere, they are exploited for thousands of applications in a densely distributed manner. Such deployment makes WSNs one of the highly anticipated key contributors of the big data nowadays. However, such big data applications raise two problems: energy consumption and user decision. First, the sensing of big data volume leads to a great waste of sensors energy, which is usually limited and not rechargeable, thus decreases the network lifetime. Second, it is a complicated mission for decision makers when dealing with a big amount of sensed data, that mostly contains a high redundancy level, to make the right decisions. In order to handle these problems, researchers have focused on the data aggregation methods in WSNs. The main goal of these methods is to minimize the huge amount of data generated by neighboring nodes thus conserving network energy and providing a useful information for the end user.

In this chapter, we consider, again, a cluster-based periodic sensor network (CPSN), where each sensor monitors the given phenomenon and periodically sends its collected data to its CH. Then, we introduce a complete data aggregation framework for CPSN. Two layer technique is proposed: at the node level and at the CH level. Our technique aims at optimizing the volume of transmitted data thus saving energy consumption and reducing bandwidth on the network level. At the first level, an aggregation process aggregates data on a periodic basis avoiding each sensor node to send its raw data to the sink. At the second level, our technique allows a cluster-head (CH) to eliminate redundant data sets generated by neighboring nodes by applying distance functions, such as Euclidean and Cosine distances. To evaluate the performance of the proposed technique, simulations on real sensor data have been conducted. Compared to other existing techniques, results show that our protocol can efficiently be used to reduce data transmission and increase network lifetime, while still keeping data integrity of the collected data.

The remainder of this chapter is organized as follows. Section 4.2 describes some of the existing data aggregation techniques proposed for WSNs. Section 4.3 describes the aggregation at the sensor level. Section 4.4 presents the aggregation method at the CH level based on the distance functions. Section 4.5 details the simulations we have conducted in real sensors data with discussion of obtained results. Finally, we conclude the chapter in Section 4.6.

4.2/ Data Aggregation: A Background

In WSNs, the performance of any data aggregation technique strongly depends on the network's topology. Hence, researchers have proposed many network's topologies for WSNs, such as Tree-based [153], Cluster-based [121], Chain-based [4] or structure free-based [32] topology.

In [70, 175, 133], the authors study the aggregation of data generated by the sensors based on a clustering topology. The authors in [70] propose EBDSC, a distributed Energy-Balanced Dominating Set-based Clustering scheme, to prolong the network lifetime by balancing energy consumption among different nodes. In EBDSC, a node becomes a candidate cluster head if it has the longest lifetime among its neighbors. In [41], the authors propose a data aggregation scheme named DMLDA, Dynamical Message List based Data Aggregation, based on clustering routing algorithm. DMLDA mainly defines a special list structure to store history messages, which is used to judge the message redundancy instead of the period delay. In [50], the authors propose to design a fuzzy based clustering and aggregation technique dedicated to under water sensor networks. In this technique the residual energy, distance to sink, node density, load and link quality are parameters considered as inputs to the fuzzy logic. Based on the output of fuzzy logic module, appropriate cluster heads are elected and act as aggregator nodes. The authors in [129] propose an Under Water Density Based Clustered Sensor Network (UWDBCSN) scheme using heterogeneous sensors. The method helps in reducing overall communication costs, in electing the cluster-head, it also increases the overall network lifetime.

Other proposed techniques of data aggregation are based on a tree network topology, such as [105, 85, 172]. The authors in [105] use Genetic Algorithm (GA) to calculate all possible routes represented by the aggregation tree. The objective is to find the optimum tree which is able to balance the data load and the energy in the network. In [85], a semi-structured protocol based on the multi-objective tree is proposed, in order to reduce transmission delays and enhance the aggregation probability. In such a work, the routing scheme explores the optimal structure by using the Ant Colony Optimization (ACO). The authors in [43, 114] propose Tree on DAG (ToD) technique, a semi-structured approach that uses Dynamic Forwarding on an implicitly constructed structure composed of multiple shortest path trees to support network scalability. The key principle behind ToD was that adjacent nodes in a graph will have low stretch in one of these trees in ToD, thus resulting in early aggregation of packets. In [172], the authors propose a method to build an aggregation tree model in WSN such that the captured data is aggregated along the route from the leaf cells to the root of the tree. In this scheme, the tree is not built directly on sensors, but on the non-overlapping cells, which are divided with equal sizes in the target terrain. A representative sensor in each cell acts in name of the whole cell, including forwarding and aggregation of the sensing data in its cell and the receiving data from the neighbor cells.

Other works on data aggregation in WSNs are based on a chain routing topology [35, 62, 126]. In [35], the authors propose a Cycle-Based Data Aggregation Scheme (CBDAS) in order to reduce the amount of data transmitted to the base station (BS). In CBDAS, the network is divided into a grid of cells, each with a head. The network lifetime is prolonged by linking all cell heads together to form a cyclic chain, where the gathered data move from node to node along the chain, getting aggregated. In [62], a chain-based routing scheme for application-oriented cylindrical networks is proposed. After finding local optimum paths in separate chains at each scheme, the authors formulate mathematical models to find a global optimum path for data transmission through their interconnection.

Finally, some works proposed recently on data aggregation are based on a structure-free of the network [32, 86]. In [32], the authors propose a Structure-Free and Energy-Balanced data aggregation protocol, SFEB. SFEB features both efficient data gathering and balanced energy consumption, which result from its two-phase aggregation process and the dynamic aggregator selection mechanism. In [86], a virtual force-based dynamic routing algorithm (VFE) for data aggregation in WSNs is proposed. Motivated by the cost field and virtual force theories, VFE allows each node to select the optimal node to be the next hop which makes data aggregation more efficient. The authors in [135] propose a data aggregation with multiple sinks in an Information-Centric Wireless Sensor Network with an ID-based information-centric network, in order to reduce the energy-transmission cost.

Subsequently, clustering is recently considered as an efficient topology control method in WSN [15] that has many advantages, especially as far as scalability and network maintenance are concerned, compared to other topologies. However, most of the existing data aggregation techniques based on clustering topology are dedicated to event driven data model [6] and they mainly focus on the selection of CHs [100, 72]. In these techniques only CHs process and aggregate data without any processing at the level of the nodes themselves. Consequently, as explained in Chapter 4, PFF technique [22] can be considered as one of the most efficient data aggregation technique proposed for cluster-based PSN because it performs aggregation at both levels (sensors and CH). Hence, in this chapter, we compare the results of our proposed technique to those obtained in PFF.

4.3/ AGGREGATION AT SENSOR LEVEL

Since the $Similar$ function defined in Chapter 3 has been proven as an efficient method to eliminate redundant data collected at each period, we propose to use it in our technique as an aggregation method at the sensor level. Hence, based on the

$Similar$ function 3.1 and the Algorithm 8, each sensor S_i will have a set of readings, R'_i={$(r'_{i_1}, wgt(r'_{i_1})), (r'_{i_2}, wgt(r'_{i_2})), \ldots, (r'_{i_k}, wgt(r'_{i_k}))$}, with no redundant values. In addition, we reuse the notations $|R'_i|$ 3.3 and $wgt_c(R'_i)$ 3.4 to express the cardinality and the weighted cardinality of the set R'_i.

Therefore, at the end of this aggregation level, each sensor node S_i will send its set R'_i to the CH at the end of each period. In the next section, we present the aggregation method at the CH level which, in turn, aggregates the data sets coming from different member nodes.

4.4/ Aggregation at CH Level

In this section, we propose an aggregation method to search redundant data sets generated by the sensors using the distance functions. Distance functions are an important method that can find duplicated data sets by searching dissimilarities between these sets. Hence, a great number of distance functions have been proposed in the literature [38]. In this chapter, we are interested in two distance functions that are widely used in various domains: Euclidean and Cosine distances.

Let us consider two data sets R'_i and R'_j, generated by the sensor nodes S_i and S_j respectively, at the period p as follows: $R'_i = \{(r'_{i_1}, wgt(r'_{i_1})), (r'_{i_2}, wgt(r'_{i_2})), \ldots, (r'_{i_{k_i}}, wgt(r'_{i_{k_i}}))\}$ and $R'_j = \{(r'_{j_1}, wgt(r'_{j_1})), (r'_{j_2}, wgt(r'_{j_2})), \ldots, (r'_{j_{k_j}}, wgt(r'_{j_{k_j}}))\}$ where $|R'_i| = k_i$ and $|R'_j| = k_j$. Therefore, R'_i and R'_j are considered redundant if the calculated distance between them is less than a threshold (t_d) as follows:

$$Dist(R'_i, R'_j) \leq t_d$$

However, two issues must be considered when using distance functions with readings weights: **1)** Calculating the distance between two data sets with different cardinalities, e.g. k_i and k_j, and **2)** integrating the weights when calculating the distance between sets. To overcome these challenges, we propose to use the threshold δ, introduced in the $Similar$ function (cf. Section 3.3), when computing the distance between the sets.

In order to find the distance between two sets R'_i and R'_j, the first step consists in dividing each set into two parts: overlap and remained. The overlap part of the set R'_i (resp. R'_j) contains readings that are similar to those in R'_j (resp. R'_i) while the remained part contains the remaining readings of R'_i (resp. R'_j). Subsequently, the overlap part between two sets has already been defined in Definition 3.5, i.e. $R'_i \cap_s R'_j$, while the remained part in each set is defined as follows:

Definition 4.1 *Remained part of R'_i, R'_{i_r}. Consider two sets of sensor readings R'_i and R'_j. We define the remained part R'_{i_r} (respectively R'_{j_r}) as all the readings in R'_i (respectively R'_j) minus the readings in the overlap part of R'_i (respectively R'_j) as follows:*

$$\begin{cases} R'_{i_r} & = & R'_i \ominus (R'_i \cap_s R'_j) \\ and & & \\ R'_{j_r} & = & R'_j \ominus (R'_i \cap_s R'_j) \end{cases}$$

Where $\ominus$ is a new operation defined as:

Definition 4.2 *Minus Operation, $\ominus$. We define the minus operation, $R'_i \ominus R'_j$, between two sets R'_i and R'_j as all the readings in R'_i and not in R'_j as follows:*
$R'_i \ominus R'_j = \{r'_i \in R'_i,$ *with* $wgt(r'_i) = wgt(r'_i) - wgt(r'_j)$ *for all* $r'_j \in R'_i \cap_s R'_j$ *and* $Similar(r'_i, r'_j) = 1\}$

In order to compute the distance between R'_i and R'_j, we must transform R'_{i_r} (respectively R'_{j_r}) into a vector as follows:

$$vR'_{i_r} = [\, \underbrace{r'_{i_1}, \ldots, r'_{i_1}}_{wgt(r'_{i_1})\text{ times}}, \underbrace{r'_{i_2}, \ldots, r'_{i_2}}_{wgt(r'_{i_2})\text{ times}}, \ldots, \underbrace{r'_{i_{k_i}}, \ldots, r'_{i_{k_i}}}_{wgt(r'_{i_{k_i}})\text{ times}} \,]$$

Then, we order the readings in vR'_{i_r} (respectively vR'_{j_r}) by increasing order of their values to ensure a logical comparison when calculating the distance between them.

4.4.1/ Euclidean Distance

In mathematics, the Euclidean distance is the ordinary distance, e.g. straight line distance, between two points, sets or objects. It is used in many applications and domains, such as computer vision and prevention of identity theft [107]. Furthermore, the Euclidean distance is already used in WSN during the deployment phase in terms of sensors' localization [7] and inter-sensors distance estimations [131]. In this paper, we use the Euclidean distance on the data sets collected by sensors while adapting it to take into account the measures' weights.

In general, the Euclidian distance (E_d) between two data sets R_i and R_j, before applying the $Similar$ function, is given by:

$$E_d(R_i, R_j) = \sqrt{\sum_{k=1}^{\mathcal{T}} (r_{i_k} - r_{j_k})^2} \text{ where } r_{i_k} \in R_i \text{ and } r_{j_k} \in R_j$$

Thus, R_i and R_j are said to be redundant if $E_d(R_i, R_j) \leq t_d$, where t_d is a threshold determined by the application.

After applying the $Similar$ function, we consider that R_i and R_j are respectively transformed into R'_i and R'_j. Therefore, we calculate the Euclidean distance between R'_i and R'_j as follows:

$$E_d(R'_i, R'_j) = \sqrt{\sum_{k=1}^{|vR'_{i_r}|} (r'_{i_k} - r'_{j_k})^2} \text{ where } r'_{i_k} \in vR'_{i_r} \text{ and } r'_{j_k} \in vR'_{j_r} \qquad (4.1)$$

Proof. Consider two sets of data R'_i and R'_j. Then:

$$\begin{aligned}
E_d(R'_i, R'_j) &= \sqrt{(R'_i - R'_j)^2} \\
&= \sqrt{\left((R'_i \cap_s R'_j + vR'_{i_r}) - (R'_i \cap_s R'_j + vR'_{j_r})\right)^2} \\
&= \sqrt{\left((R'_i \cap_s R'_j - R'_i \cap_s R'_j) + (vR'_{i_r} - vR'_{j_r})\right)^2} \\
&= \sqrt{(vR'_{i_r} - vR'_{j_r})^2} \\
&= \sqrt{\textstyle\sum_{k=1}^{|vR'_{i_r}|} (r'_{i_k} - r'_{j_k})^2} \quad \text{where } r'_{i_k} \in vR'_{i_r} \text{ and } r'_{j_k} \in vR'_{j_r}
\end{aligned}$$

□

In the above proof, we consider that the Euclidean distance between the readings in the overlap is equal to zero because they are considered redundant at the sink. Therefore, the Euclidean distance between two sets is equal only to distance between readings in the remained parts of R'_i and R'_j, i.e. vR'_{i_r} and vR'_{j_r} respectively.

4.4.2/ Cosine Distance

Cosine distance is a measure of dissimilarity between two vectors that measures the cosine of the angle between them. This kind of dissimilarity has been used widely in many aspects, such as the anomaly detection in web documents [44] and medical diagnosis [168]. Depending on the angle between the vectors, the resulting dissimilarity ranges from -1 meaning exactly the opposite, to 1 meaning exactly the same.

The Cosine distance (C_d) between two sets R_i and R_j, before applying $Similar$ function, is given by:

$$C_d(R_i, R_j) = 1 - \frac{\sum_{k=1}^{\mathcal{T}} (r_{i_k} \times r_{j_k})}{\sqrt{\sum_{k=1}^{\mathcal{T}} r_{i_k}^2} \times \sqrt{\sum_{k=1}^{\mathcal{T}} r_{j_k}^2}} \quad \text{where } r_{i_k} \in R_i \text{ and } r_{j_k} \in R_j.$$

Thus, R_i and R_j are redundant if $C_d(R_i, R_j) \leq t_d$.

Then, we adapt the Cosine distance to the readings weights in R'_i and R'_j as follows:

$$C_d(R'_i, R'_j) = 1 - \frac{A + \sum_{k=1}^{|vR'_{i_r}|} (r'_{i_{rk}} \times r'_{j_{rk}})}{\sqrt{A + \sum_{k=1}^{|vR'_{i_r}|} r'^2_{i_{rk}}} \times \sqrt{A + \sum_{k=1}^{|vR'_{j_r}|} r'^2_{j_{rk}}}} \tag{4.2}$$

$$\text{where } A = \sum_{k=1}^{|R'_i \cap_s R'_j|} (wgt_{min}(r'_{i_k}, r'_{j_k}) \times r'^2_{i_k})$$

Proof. Consider two sets of data R'_i and R'_j. Then:

$$
\begin{aligned}
C_d(R'_i, R'_j) &= 1 - \frac{R'_i \times R'_j}{\sqrt{R'^2_i} \times \sqrt{R'^2_j}} \\
&= 1 - \frac{\left(R'_i \cap_s R'_j + vR'_{i_r}\right) \times \left(R'_i \cap_s R'_j + vR'_{j_r}\right)}{\sqrt{(R'_i \cap_s R'_j)^2 + vR'^2_{i_r}} \times \sqrt{(R'_i \cap_s R'_j)^2 + vR'^2_{j_r}}} \\
&= 1 - \frac{(R'_i \cap_s R'_j)^2 + (vR'_{i_r} \times vR'_{j_r})}{\sqrt{(R'_i \cap_s R'_j)^2 + vR'^2_{i_r}} \times \sqrt{(R'_i \cap_s R'_j)^2 + vR'^2_{j_r}}} \\
&= 1 - \frac{A + \sum_{k=1}^{|vR'_{i_r}|} (r'_{i_{rk}} \times r'_{j_{rk}})}{\sqrt{A + \sum_{k=1}^{|vR'_{i_r}|} r'^2_{i_{rk}}} \times \sqrt{A + \sum_{k=1}^{|vR'_{j_r}|} r'^2_{j_{rk}}}}
\end{aligned}
$$

where $A = \sum_{k=1}^{|R'_i \cap_s R'_j|} (wgt_{min}(r'_{i_k}, r'_{j_k}) \times r'^2_{i_k})$

□

4.4.3/ Distance Normalization

In general, each distance function has its own method to calculate the distance between data sets. For instance, straight-line distance in Euclidean distance and the angle between data sets in Cosine distance. Therefore, normalization becomes essential to scale the distance between data sets into the range $[0, 1]$ to have thus the same variation between sets before comparing them. Many researches have been conducted on vector normalization in different domains [117, 108]. However, some of such methods are dedicated to particular applications where others scale data sets into a very narrow range [108]. Otherwise, the Gaussian normalization is a commonly used method to normalize data in various domain such as WSNs. Hence, in this chapter, data sets sent by the sensor nodes to the CH are normalized using Gaussian normalization. Once the CH receives the data sets at each period, it calculates first the distance, Euclidean or Cosine, for each pair of sets as follows: $d = \{d_1(R'_1, R'_2), d_2(R'_1, R'_3), \ldots, d_{\frac{n\times(n-1)}{2}}(R'_{n-1}, R'_n)\}$ where n is the number of total sets and $\frac{n\times(n-1)}{2}$ is the number of all possible distances. Then, it normalizes the returned distance values using the following Gaussian normalization equation:

$$
d'_i = \frac{d_i - \overline{Y}}{6 \times \sigma} + \frac{1}{2} \tag{4.3}
$$

where $\overline{Y}$ is the mean of all distances and σ is the standard deviation of pairwise distance over all data. $\overline{Y}$ and σ are calculated as follows:

$$
\overline{Y} = \frac{\sum_{k=1}^{|d|} d_k}{|d|} \quad \text{and} \quad \sigma = \sqrt{\frac{\sum_{k=1}^{|d|} (d_i - \overline{Y})^2}{|d|}}, \text{ where } |d| = \frac{n \times (n-1)}{2}
$$

After normalizing all pairwise distances, the CH will form the distance normalization vector between each pair of sets as follows: $d' = \{d'_1(R'_1, R'_2), d'_2(R'_1, R'_3), \ldots, d'_{\frac{n\times(n-1)}{2}}(R'_{n-1}, R'_n)\}$.

4.4.4/ Distance-based Algorithm at the CH Level

In this section we present our data aggregation method at the CH based on the distance functions. Algorithm 8 describes how the CH finds redundant sets of readings generated by sensors then how it selects, among them, data sets to be sent to the sink. After having normalized data sets based on Equation 4.3 (lines 2 to 11), the CH considers that two sets are redundant if the normalized distance between them is less than the threshold t_d (line 12 and 13). The $Dist$ function in line 5 represents Euclidean or Cosine distances and can be calculated based on Equations 4.1 and 4.2 respectively. Then, for each pair of redundant set, the CH chooses the one having the highest cardinality (line 18), then it adds it to the list of sets to be sent to the sink (line 19). After that, it removes all pairs of redundant sets that contain R'_i or R'_j from the set of pairs (which means it will not check them again). Finally, the CH assigns to each set its weight (line 21) when sending it to the sink.

Algorithm 8: Distance-based Redundancy Searching Algorithm

Data: Set of readings' sets $R' = \{R'_1, R'_2 \ldots R'_n\}$, t_d
Result: List of sent sets, L

```
1  $S \leftarrow \emptyset$;
2  $d \leftarrow \emptyset$ // list of pairwise distance;
3  for each set $R'_i \in R'$ do
4      for each set $R'_j \in R'$ such that $R'_j \neq R'_i$ do
5          compute $Dist(R'_i, R'_j)$;
6          $d \leftarrow d \cup \{Dist(R'_i, R'_j)\}$;
7      end
8  end
9  compute $\overline{Y}$ and $\sigma$ for $d$;
10 for each $d_i \in d$ do
11     $d'_i = ((d_i - \overline{Y})/(6 \times \sigma)) + 0.5$;
12     if $d'_i \leq t_d$ then
13         $S \leftarrow S \cup \{(R'_i, R'_j)\}$;
14     end
15 end
16 $L \leftarrow \emptyset$;
17 for each pair of sets $(R'_i, R'_j) \in S$ do
18     Consider $|R'_i| \geq |R'_j|$;
19     $L \leftarrow L \cup \{R'_i\}$;
20     Remove all pairs of sets containing one of the two sets $R'_i$ and $R'_j$;
21     $wgt(R'_i)$ = number of removed pairs + 1;
22 end
23 return $L$;
```

4.5/ Simulation Results and Evaluation

In this section, we present the simulation results which evaluate the performance of our proposed technique. The objective of these simulations is to confirm that the proposed data aggregation method can successfully achieve desirable results for energy conservation, data latency and data accuracy in different monitoring applications. Again, we used the publicly available Intel Lab dataset which contains data collected from 46 sensors deployed in the Intel Berkeley Research Lab [88][1]. For the sake of simplicity, in this chapter we are interested in one field of sensor readings: the temperature. Similarly to simulations in Chapter 3, we assume that all nodes send their data to a common CH placed at the center of the Lab. First, each node periodically reads real readings while applying the $Similar$ function. At the end of this step, each node sends its set of readings with weights to the CH which in its turn aggregates them using the proposed aggregation method in our technique. Furthermore, we compare the results of our technique to those of the PFF technique proposed in [22]. We have implemented both techniques on a Java simulator and we compared the results of 15 periods in all the experiments.

We evaluated the performance using the following parameters:

- **(a)** the threshold δ, defined in $Similar$ function, takes the following values: 0.03, 0.05, 0.07, 0.1.
- **(b)** $\mathcal{T}$, the number of sensor readings taken by each sensor node during a period, takes the following values: 200, 500 and 1000.
- **(c)** the distance threshold t_d takes the following values: 0.35, 0.4, 0.45 and 0.5.
- **(d)** the threshold t of the Jaccard similarity function in PFF technique is fixed to 0.75.

4.5.1/ Data Aggregation Ratio at Sensor Level

During the aggregation at sensor level (or aggregation node phase), each sensor node searches the similarity between readings captured at each period and assigns for each measure its weight. Figure 4.1 shows the percentage of remaining data, or aggregated data, which will be sent to the CH, with and without applying the aggregation node phase at the sensors level. At each period, the amount of data collected by each sensor is reduced at least by 77% (and up to 94%) after applying the $Similar$ function. Otherwise, the sensor node sends all the collected data, e.g. 100%, without applying the aggregation node phase. Therefore, our technique can successfully eliminate redundant measures at each period and reduce the amount of data sent to the CH. We can also observe that, with the aggregation node phase at sensor level, data redundancy among data increases when $\mathcal{T}$ or δ increases.

[1] Our technique has been also applied on real data collected from the ARGO project [115]. The obtained results were similar to those presented in this chapter which indicate the efficiency of our technique in underwater sensor applications. However, the results are not presented in order to not increase the number of pages of this chapter.

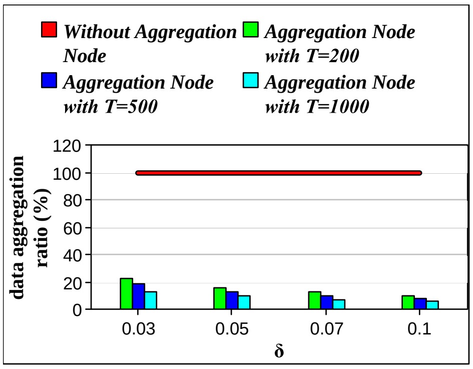
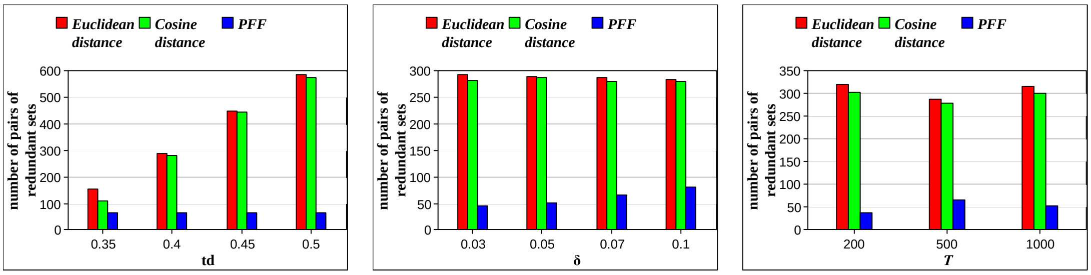


Figure 4.1: Percentage of data after applying aggregation node level.

4.5.2/ Data Sets Redundancy

When receiving all the sets from its member nodes at the end of each period, CH applies the second aggregation level in order to find all pairs of redundant sets. Figure 4.2 shows the number of pairs of redundant sets obtained at each period when applying Euclidean and Cosine distances and PFF technique. First, we fixed $\mathcal{T}$ and δ and we varied t_d as shown in Figure 4.2(a), then we fixed $\mathcal{T}$ and t_d and varied δ as shown in Figure 4.2(b) and, finally, we fixed δ and t_d and varied $\mathcal{T}$ as shown in Figure 4.2(c). The obtained results show that, the CH finds more redundant sets when applying the distance functions, i.e. Euclidean and Cosine, compared to the Jaccard function used in PFF technique for different values of parameters. This is because the distance condition (Equations 4.1 and 4.2) is more flexible compared to the Jaccard similarity condition used in PFF.



(a) $\mathcal{T}$ = 500, δ=0.07. (b) $\mathcal{T}$ = 500, t_d=0.4. (c) δ=0.07, t_d=0.4.

Figure 4.2: Number of pairs of redundant sets at each period.

Several observations can be made based on the obtained results in Figure 4.2:

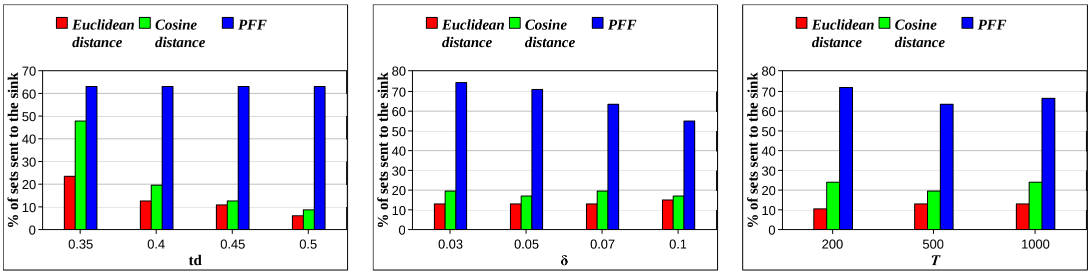
- The Euclidean distance finds more redundant sets compared to Cosine distance in all cases. This is due to the Euclidean distance equation which is more flexible function compared with that used in Cosine distance.

- The number of pairs of redundant sets in Euclidean and Cosine distances increases when t_d increases (Figure 4.2(a)). This is because, we allow more measures to be eliminated when t_d increases thus we allow more sets to be redundant.

- The number of redundant sets obtained in both distances is almost fix when fixing $\mathcal{T}$ and t_d and increasing δ, while it increases in PFF (Figure 4.2(b)). This is because the data sets save the same distance condition when changing δ. Otherwise, the results of PFF proportionally change with δ since they are strongly dependent on the $Similar$ function.

4.5.3/ Data Sets Reduction

In this section, we show how the CH is able to eliminate redundant sets at each period before sending them to the sink. In other words, how many sets among the redundant sets the CH will send to the sink at each period (lines 16-23 in Algorithm 8). Figure 4.3 shows the percentage of the remaining sets that will be sent to the sink after eliminating the redundancy. Similarly to Figure 4.2, we varied t_d and we fixed $\mathcal{T}$ and δ in Figure 4.3(a), then we varied δ and fixed $\mathcal{T}$ and t_d in Figure 4.3(b) and, finally, we varied $\mathcal{T}$ and fixed δ and t_d in Figure 4.3(c). Generally, the obtained results are dependent on the number of the redundant sets shown in Figure 4.2; if more redundant sets are found, this will lead to more sets being eliminated. Therefore, Euclidean and Cosine distances allow the CH to eliminate more redundant sets at each period compared to PFF technique, except when t_d is small (e.g. $t_d = 0.35$ in Figure 4.3(a)).



(a) $\mathcal{T} = 500$, δ=0.07.

(b) $\mathcal{T} = 500$, t_d=0.4.

(c) δ=0.07, t_d=0.4.

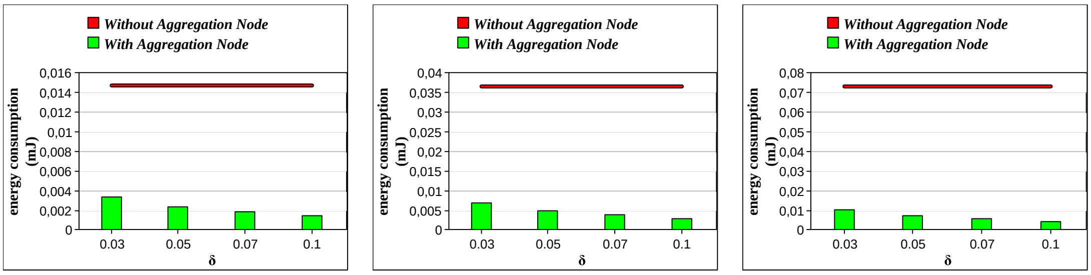
Figure 4.3: Percentage of sets sent to the sink at each period.

The results obtained in Figure 4.2 allow us to conclude some observations shown in Figure 4.3:

- The percentage of sets sent to the sink using Euclidean distance is inferior to that sent using Cosine distance, for different values of parameters. This is because the CH finds more redundant sets by using Euclidean distance (see results in Figure 4.2).
- The distance functions allow the CH to send 15% to 61% less sets to the sink compared to PFF, due to the flexibility of distances regarding the redundancy compared to similarity functions.
- The percentage of sets sent to the sink in Euclidean and Cosine distances decreases when t_d increases (Figure 4.3(a)) while it is almost fix when δ or $\mathcal{T}$ increases (Figure 4.3(b) and 4.3(c)).

4.5.4/ Energy Consumption Study

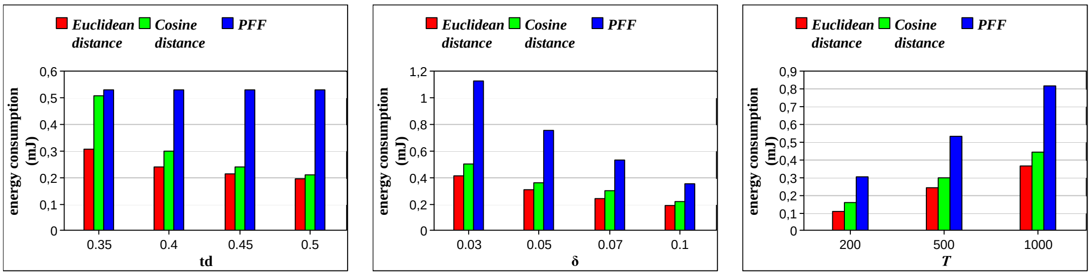
In this section, our objective is to study the energy consumption at the sensor nodes and CH levels. Recall, we always use the same radio model as discussed in [88] to evaluate the energy consumption. In sensor networks, energy consumption is highly dependent on the amount of data sent and received. First, Figure 4.4 shows the energy consumption comparison with and without applying the aggregation node phase by each sensor node and when varying $\mathcal{T}$ and δ. Since the aggregation node significantly reduces the redundancy among data collected by the sensor node (see results in Figure 4.1), it allows it to proportionally save its energy when transmitting its data to the CH at each period. This result is obvious in Figure 4.4 when the sensor node applies the aggregation node phase and when δ or $\mathcal{T}$ increases. It is important to notice that our technique can save from 76% (Figure 4.4(a)) up to 94% (Figure 4.4(c)) of the energy of a sensor node.



(a) $\mathcal{T}$ = 200. (b) $\mathcal{T}$ = 500. (c) $\mathcal{T}$ = 1000.

Figure 4.4: Energy consumption in each sensor node.

Figure 4.5 shows the energy consumption comparison at the CH when using distances method and PFF technique, in function of t_d in Figure 4.5(a), of δ in Figure 4.5(b) and of $\mathcal{T}$ in Figure 4.5(c). Depending on the results obtained in Figure 4.3, the distances method gives the best results, except for $t_d = 0.35$ in Cosine, regarding the energy consumption in the CH. Subsequently, Euclidean distance can reduce the energy consumed in CH up to 64% compared to the amount of energy consumed using PFF. Otherwise, Cosine distance can reduce up to 60% of the energy consumption in the CH compared to that consumed using PFF.



(a) $\mathcal{T}$ = 500, δ=0.07. (b) $\mathcal{T}$ = 500, t_d=0.4. (c) δ=0.07, t_d=0.4.

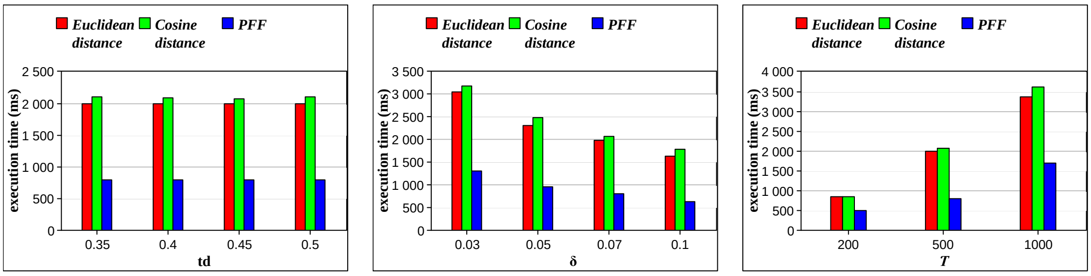
Figure 4.5: Energy consumption at the CH.

Since the energy consumption is minimized when the percentage of sets sent is minimized, several observations shown in Figure 4.5 can be concluded:

- Euclidean distance decreases the energy consumption in the CH from 9% to 40% compared to the Cosine distance. This is because the Euclidean distance sends less sets to the sink compared to the Cosine distance.
- Using Euclidean and Cosine distances, the CH conserves more energy when t_d increases (Figure 4.5(a)).
- The energy consumption in the CH using the distance functions is almost independent from δ threshold (Figure 4.5(b)). Otherwise, PFF reduces the energy consumption in the CH when δ increases (Figure 4.5(b)).

4.5.5/ Data Latency: Execution Time

In this section, we compare the execution time required for both data aggregation techniques when varying t_d, δ and $\mathcal{T}$ respectively (Figure 4.6). The execution time is dependent on the normalization process of data sets in distances method and on the number of candidates generated in PFF. The obtained results show that PFF can accelerate the execution time at the CH twice faster than distances method; the reason for that is the normalization used in Euclidean and Cosine distances which needs to calculate all distances between pairs of sets while PFF only searches the similarity between the generated candidate pairs.



(a) $\mathcal{T} = 500$, δ=0.07.

(b) $\mathcal{T} = 500$, t_d=0.4.

(c) δ=0.07, t_d=0.4.

Figure 4.6: Execution time at the CH.

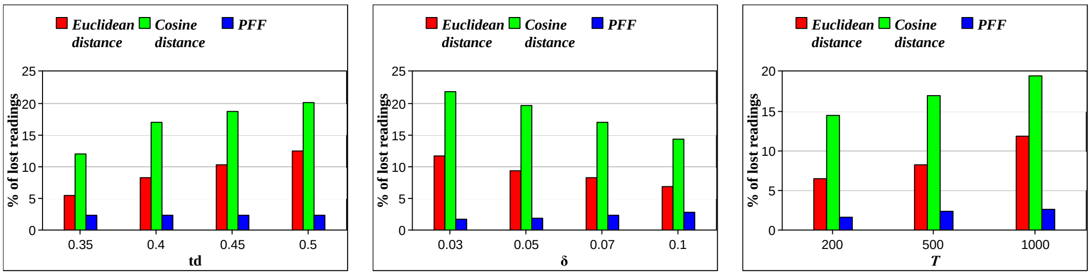
Several observations can be made based on the results shown in Figure 4.6:

- The Euclidean distance decreases the execution time at the CH more than the Cosine distance. This is due to the complexity of the calculation of Cosine distance (Equation 4.2) compared to Euclidean distance (Equation 4.1).
- The execution time required for both distances is almost fix when varying t_d (Figure 4.6(a)). This is because both distances must normalize all data sets independently from t_d value.
- The data latency at the CH is optimized when δ increases, in both techniques (Figure 4.6(b)). This is because the cardinality of a data set decreases when δ increases thus the computation between sets decreases as well.

- The CH requires, with both aggregation techniques, more execution time when $\mathcal{T}$ increases (Figure 4.6(c)). This is because the cardinality of sets increases thus requiring more time to compare these sets.

4.5.6/ Data Accuracy: Integrity of Information

Data accuracy is an important factor in WSNs which represents the measure “loss rate”. It is an evaluation of the measures taken by the sensor nodes whose values (or similar values) do not reach the sink. Figure 4.7 shows the results of data accuracy for the data aggregation functions used in our technique for different values of t_d, δ and $\mathcal{T}$. We can notice that PFF gives the best results for data accuracy, 2.81% in the worst case, compared to the Euclidean (up to 12.52%) and Cosine (up to 21.75%) distances. The reason for this is that the Jaccard function used in PFF is a strong constraint regarding the loss measures compared to distance constraint which is more flexible.



(a) $\mathcal{T}$ = 500, δ=0.07. (b) $\mathcal{T}$ = 500, t_d=0.4. (c) δ=0.07, t_d=0.4.

Figure 4.7: Data accuracy.

The following observations can be made based on the results obtained in Figure 4.7:

- The Euclidean distance conserves the integrity of data more than Cosine distance in all cases. This is due to the equation of Cosine distance which eliminates the sets that have high cardinality.
- The loss of measures using Euclidean and Cosine distances increases when t_d increases (Figure 4.7(a)). This is because the CH eliminates more sets when t_d increases (see results in Figure 4.3).
- The data accuracy in both distances increases when δ increases (Figure 4.7(b)) or $\mathcal{T}$ decreases (Figure 4.7(c)). On the other hand, using PFF, the data accuracy decreases when δ or $\mathcal{T}$ increases.

4.5.7/ Further Discussions

In this section, we give further consideration to our proposed technique compared to PFF. We give some directions as to which technique should be chosen, under which conditions and in which circumstances of the application.

From the energy preserving point of view at the CHs, both techniques significantly reduce the energy consumption in the CHs. In addition, we observe that the distance method conserves more energy compared to the similarity method. Subsequently, it reduces up to 64% of the energy in CH compared to PFF. Therefore, in the applications where we need to conserve the energy of the network as long as possible, the distance method is more suitable.

Although PFF outperforms at most to twice the execution time at the CH compared to distance functions, both techniques provide good performance regarding the data latency. Consequently, when the priority for the application is to deliver data to the sink, similarity and distance functions can be considered as suitable techniques.

From the data accuracy point of view at the CHs, the similarity method can totally save the integrity of the collected data without any loss of information, e.g. up to 2.81%. On the other hand, the distance method gives acceptable results for data accuracy. Hence, if the application does not permit flexibility regarding data accuracy, the similarity functions method is more suitable; else, distance functions can be used as a compromise between energy saving and data accuracy flexibility.

To summarize this section, Table 4.1 shows the flexibility of each technique regarding energy consumption, data latency and accuracy, and complexity of the method at the CHs.

Table 4.1: Comparison between distance and similarity functions.

Techniques	Energy consumption conserving	Data latency	Data accuracy	Complexity
Euclidean distance	very good	medium	good	O(n^2)
Cosine distance	good	medium	medium	O(n^2)
PFF	low	good	very good	O($n \times log(n)$)

4.6/ Conclusion

In this chapter, we have introduced a complete data aggregation framework for cluster-based periodic sensor networks. We proposed energy-efficient technique which are applied at each cluster separately and achieve aggregation at both sensor nodes and CH levels. Further to a local aggregation at sensor nodes level, our technique allows CHs to eliminate redundant data sets generated by neighboring nodes at each period by applying distance functions, such as Euclidean and Cosine. We have demonstrated through experiments on real data readings the efficiency of our proposed technique in sensor networks in terms of energy consumption, data latency and accuracy.

5

Spatio-Temporal Data Correlation with Scheduling Strategies

The explosive growth of the data volume generated in PSNs applications has led to one of the most challenging research issues of the big data era. To deal with such amounts of data, exploring data correlation and scheduling strategies have received great attention in sensor networks. In this chapter, we propose an efficient mechanism based on the Euclidean distance for searching the spatial-temporal correlation between sensor nodes in periodic applications. Based on this correlation, we propose two sleep/active strategies for scheduling sensors in the network. The first one searches the minimum number of active sensors based on the set covering problem while the second one takes advantages from the correlation degree and the residual energy of the sensors for scheduling them in the network.

5.1/ Introduction

Due to the tremendous growth of the information and communication technology nowadays, such as social media [163, 24], video surveillance [165, 164], cloud computing [164, 160] etc., the era of Big Data is open up. Wireless sensor networks (WSNs) can be considered as one of the most important source of big data era. In some applications, such as healthcare services and atmospheric conditions monitoring [25] and commercial flights [55], the volume of data generated by sensors nodes reaches the order of petabytes every day. Moreover, the sensor nodes have a limited energy supply and their generated data are following the 4V feature (volume, velocity, variety and value) of big data [130, 137]. Therefore, the problems of energy constraint and data redundancy emerge inevitably at the core of WSNs challenges.

To deal with big data generated in WSNs, recent studies [155, 59, 118] pay a great attention to inter-nodes data correlation techniques and scheduling nodes strategies. First, by studying the spatial-temporal correlation between sensors, the high redundancy existing in sensed data will be removed. This leads to reduce the volume of big data routed in the network thus the useful information only will be transmitted to the sink node. Second, scheduling strategies play a significant role in conserving sensors energies and extending the lifetime of WSNs. When sensor nodes are considered redundant, scheduling strategies select a subset of sensors to collect data while the remaining nodes will be scheduled to the sleep mode. Hence, the combination of inter-nodes data correlation techniques and scheduling strategies can yield a great potential in increasing the energy efficiency of WSNs and enables the efficient handling of big data applications.

In this chapter, our main goal is to minimize the huge amount of data generated in clustering-based periodic sensor networks, by searching the spatial and temporal correlation between neighboring nodes. When correlated nodes are detected, we propose two scheduling strategies in order to switch sensors in each cluster into sleep/active modes. The first strategy is based on the set cover problem while the second strategy takes into account the correlation degree and the residual energy of the sensors when scheduling nodes in the cluster.

The remainder of this paper is organized as follows. Section 5.2 gives a background about the spatio-temporal data correlation in sensor networks. Section 5.3 describes our mechanism, based on the Euclidean distance, for searching spatially-temporally correlated nodes. In Section 5.4, we propose two strategies for scheduling sensors in the network. Simulation results based on real data readings are exposed in Section 5.5. Finally, we conclude the chapter in Section 5.6.

5.2/ Data Correlation: A Background

In WSNs, exploring inter-nodes correlation is a well-known strategy which helps in increasing the battery life of sensor nodes [48, 149]. In the literature, we can distinguish between three main categories of data correlation between nodes: spatial, temporal or spatio-temporal correlations. In [28, 150], the authors give a survey about different data collection techniques proposed for each category of data correlation in WSNs.

5.2.1/ Spatial Correlation

Geographical location of sensors plays an important role in WSNs. That is, the generated sensory data by neighboring nodes are often correlated. Hence, there has been very active research in data spatial correlation in sensor networks [36, 27, 81, 132, 134].

In [134], the authors present an improved method to prolong the lifetime of WSN by exploiting the spatial correlation which is called Unequal Distributed Spatial Correlation-based Tree Clustering for Approximate Data Collection (UDSCTC) algorithm. UDSCTC changes the nodes' competition radius which makes the clusters unequal. Then it can divide the network with minimizing reading dissimilarity of nodes in the same cluster and can prolong the network lifetime by making the clusters which are nearer to the sink smaller. In [87], an α-local spatial clustering algorithm for WSNs is proposed. By measuring the spatial correlation between data sampled by different sensors, the algorithm constructs a dominating set as the sensor network backbone used to realize the data aggregation based on the information description/summarization performance of the dominators. The authors in [37] propose an enhanced version of LEACH protocol. It applies aggregation strategies in the area monitored by sensor nodes to reduce the number of reports sent to the sink and to save energy. The proposed approach seeks to exploit the spatial correlation among nodes and among clusters to assign different importance to the information aggregated and forwarded by the cluster head nodes.

5.2.2/ Temporal Correlation

Mostly, massive data captured by sensor nodes then routed in the network are highly temporally correlated. This correlation is due to the slow varying nature of the monitored phenomenon. As a result, temporal correlation can be detected at the sensor node level among its consecutive readings or at the CH level among readings collected by neighboring sensor nodes at the same time.

In [78, 76, 9], the authors search the temporal correlation at the sensor node level in order to eliminate redundant readings and to adapt the sampling rate. For instance, the authors in [9] propose three different approaches to utilize temporal correlation for efficient Compressive Sensing (CS) data gathering in WSN. The first approach uses temporal correlation to process the sensed data in a way that increases its sparsity order, which in turn reduces the required number of measurements in the sensing process. The second approach uses the temporal correlation as a prior in the reconstruction step for CS. The last approach combines several time instants into a single measurement vector, on the assumption that multiple measurement vector will be more sparse than distinct time instants measurements.

The temporal correlation in [22, 140, 20] is searched at the CH level among data generated by neighboring sensor nodes. For instance, the authors in [22] propose a data aggregation technique at the CH level dedicated for PSN: Prefix-Frequency Filtering (PFF). PFF uses similarity functions to allow CH to identify all pair of nodes generating similar data sets at each period. PFF can reduce data size by eliminating temporal data correlation, at each period, before sending necessary information to the sink.

5.2.3/ Spatio-Temporal Correlation

As sensed data are often correlated in both space and time, researchers have recently been motivated by exploring the spatio-temporal correlation between sensors when designing data gathering mechanisms (see [47, 34, 112, 80, 150]).

In [150], the authors propose an Efficient Data Collection Aware of Spatial-Temporal Correlation (EAST) for energy-aware data forwarding in WSNs. In EAST, nodes that detected the same event are dynamically grouped in correlated regions and a representative node is selected at each correlation region for observing the phenomenon, while the other nodes are switched to sleep mode. The authors in [34] develop a clustered spatial-temporal compression scheme by integrating network coding (NC) and compressed sensing (CS) for correlated data.

In other works, such as [149, 67, 113], the spatial-temporal correlation between sensors has been studied in order to schedule sensors in the network. In [149], an dYnamic and scalablE tree Aware of Spatial correlatTion (YEAST) is proposed. YEAST takes advantage of the best WSN routing techniques to perform energy-aware data forwarding where the entire region of sensors per event is effectively a set of representative nodes performing the task of data collection. In [67], the authors propose a centralized algorithm design and an optimizing protocol for scheduling the sensors during a specified network lifetime. The objective is to maximize the spatial-temporal coverage by scheduling sensors activity after they have been deployed.

5.2.4/ Pearson Product-Moment Coefficient (PPMC) Technique [39]

More recently, the authors in [39] propose a spatial-temporal model to extend the network lifetime based on three similarity metrics: Euclidean Distance, Cosine Similarity and Pearson Product-Moment Coefficient (PPMC). Then, based on scheduling algorithm, correlated nodes are switched to the sleep mode in order to save network energy. By performing real experiments, the authors show that PPMC metric gives better results, in terms of conserving overall energy, compared to other similarity metrics. However, PPMC has several disadvantages: **1)** it does not search the temporal correlation at the sensor node level. **2)** it does not take into account the residual energy of the sensors when switching them to the sleep mode. **3)** it assumes that all the correlated sensors have the same degree of correlation. Hence, aiming to overcome these disadvantages, we propose, in this chapter, an energy-aware spatio-temporal data collection technique based on the Euclidean distance in order to search inter-node data correlation. Once high correlation between nodes is noticed, we propose two sleep/active strategies for scheduling sensors in the network. Through simulation, we will show that our mechanism, with the two proposed strategies, can significantly outperform PPMC in terms of saving the sensors energies and extending the network lifetime.

5.3/ Spatial-Temporal Correlation Mechanism

In WSN, sensors are deployed densely in order to monitor some phenomenon which leads to have high spatial-temporal correlation between sensed data. On the one hand, sensed data are spatially correlated since the nodes are geographically close, i.e. they detect similar information. In the other hand, the temporal correlation happens due to the nature of the monitored phenomenon. Consequently, it is likely that a sensor node collects very similar data readings during a period, or, neighboring nodes generate similar data sets in the same period. In the following of this chapter, we propose a new mechanism, based on the Euclidean distance, in order to exploit spatial-temporal correlation between sensed data in WSN. Then, we propose two scheduling strategies to switch a set of sensor nodes to the sleep mode when high correlation between them is detected.

5.3.1/ Local Temporal Correlation

In periodic applications, each sensor node collects a vector of readings in each period then it send it to the CH at the end of the period. Mostly, consecutive readings collected from the sensor, in each period, are temporally correlated depending on how the monitored condition varies. We call this correlation a *local temporal correlation*. For example, in a period of one hour, the temperature sensed at each minute may not change significantly. In this case, searching local temporal correlation is necessary in order to reduce the number of reported readings and to save energy consumption in the sensor.

Let us consider a vector of readings R_i collected by the sensor S_i during period p as follows: $R_i = [r_1, r_2, \ldots, r_{\mathcal{T}-1}, r_{\mathcal{T}}]$ where $\mathcal{T}$ is the total number of readings captured during p. Thus, our objective is to explore temporal correlation between readings in R_i in order to reduce the amount of data readings that need to be transmitted and thus to save the energy in S_i. However, the $Similar$ function defined in Chapiter 3 searches similar readings in R_i without taking into account the order of these readings in the period. In this section, we propose the "$LocTmp$" function to search the similarity between *consecutive* readings in R_i. Consequently, "$LocTmp$" will save the temporal information of readings in each period for applications that need to save the readings order. Therefore, "$LocTmp$" identifies if two consecutive readings r_t and r_{t+1}, captured by the sensor S_i during a period p, are similar or not. $LocTmp$ function is defined as follows:

Definition 5.1 *$LocTmp$ function. We define the $LocTmp$ function between two consecutive readings r_t and r_{t+1} as:*

$$LocTmp(r_t, r_{t+1}) = \begin{cases} 1 & \text{if} \quad |r_t - r_{t+1}| \leq \delta, \\ 0 & \text{otherwise.} \end{cases}$$

where the value of δ is a user defined threshold and it depends on the application. Two consecutive readings captured by a sensor are considered similar if and only if their $LocTmp$ function is equal to 1.

Therefore, the $LocTmp$ function runs by each sensor node in each period in the following manner: for each new reading r_t, a sensor node S_i searches for similarity of r_t with the previous reading r_{t-1}. If r_t and r_{t-1} are similar, S_i deletes the new reading r_t and increments the weight of r_{t-1} by 1, else it adds r_t to the set and initializes its weight to 1. After searching local temporal correlation, S_i will transform the initial vector of readings, R_i, to a set of readings, R'_i, associated to their corresponding weights as follows: R'_i=$\{(r'_1, wgt(r'_1)),$ $(r'_2, wgt(r'_2)), \ldots, (r'_k, wgt(r'_k))\}$, where $k \leq \mathcal{T}$. Finally, the notations of cardinality, $|R'_i|$ 3.3 and weighted cardinality, $wgt_c(R'_i)$ 3.4, are also respected in this chapter.

5.3.2/ Spatial Correlation Between Sensors

In WSNs, satisfactory coverage of sensing area is one of the key challenges that requires a high density of sensors deployment. Environmental monitoring and military usage are good examples of typical applications in WSNs that are needed to a huge number of sensors to collect data about the surroundings, then, to send data toward the sink node. Due to such dense deployment, data sensed by the sensor nodes are spatially correlated. In addition, the closer geographically the sensors are, the higher the spatial correlation will be between their collected data. Hence, it is important to exploit the spatial correlation of data in sensor network in order to reduce the energy consumption in sensors, while conserving the integrity of these data.

Mostly, a sensor node S_i is represented by its position (x_i,y_i), its sensing range (S_r) and its transmission range (T_r). In this work, we assume that all sensor nodes have the same sensing and transmission range. Then, we use the Euclidean distance (E_g) to calculate the geographical distance between two nodes S_i and S_j as follows:

$$E_g(S_i, S_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

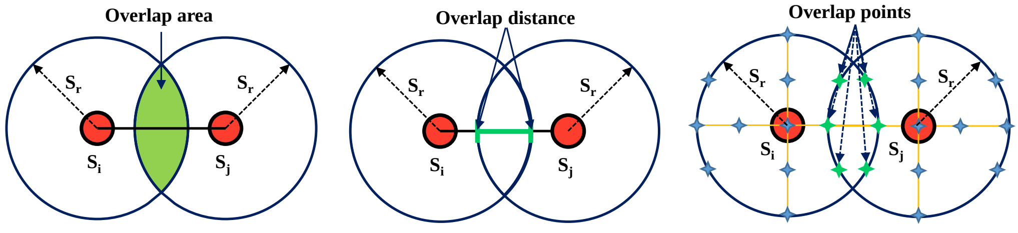
After that, we define the neighboring nodes of S_i:

Definition 5.2 *neighbor. S_j is a neighbor node of S_i if the Euclidean distance between S_i and S_j is less than the twice of sensing range as follows:*

$$E_g(S_i, S_j) \leq 2 \times S_r$$

Finally, we assume that V_i is the set of neighbors of S_i.

Subsequently, the spatial correlation between two neighboring nodes increases when the distance between them decreases. Hence, there are three main categories to search the spatial correlation between neighboring sensors. The first category, as in [79, 67], exploits the overlap area between two sensor nodes (Figure 5.1(a)). The second category, as in [144], calculates the spatial correlation based on the distance overlap between the sensors (Figure 5.1(b)). The last category, as in [61], defines a number of primary points in the circle disk of the sensing range, then it calculates the number of points in the common area between the two sensors (Figure 5.1(c)). In this chapter, we focus on the second category of spatial correlation which is simple and more flexible compared to other categories.



(a) Overlap area searching.

(b) Overlap distance searching.

(c) Overlap points searching.

Figure 5.1: Spatial correlation techniques between two sensors.

Most of the proposed studies in the literature consider that two sensors are spatially correlated if there is an overlap between their sensing range, i.e. $E_g(S_i, S_j) \leq 2 \times S_r$ in Definition 5.2. However, in our mechanism, we define a spatial correlation threshold, C_{sp}, in order to make the constraint for the spatial correlation between sensors more difficult to satisfy. Therefore, we define the spatial correlation between two sensors as follows:

Definition 5.3 *Spatial correlation between two sensors. Two given sensors S_i and S_j, where S_j is a neighbor node of S_i (i.e. $S_j \in V_i$), are spatially correlated if and only if:*

$$E_g(S_i, S_j) \leq C_{sp} \tag{5.1}$$

where C_{sp} is a threshold determined by the application and it takes values in $[0, 2 \times S_r]$. Then, we assume that V'_i is the set of all spatially correlated nodes with S_i.

Based on the Definition 5.3, we can dynamically change the threshold C_{sp} depending on the criticality of the monitored environmental; if the phenomenon is critical, the decision makers can decrease C_{sp} in order to decrease the number of spatially correlated nodes for each node, i.e. $|V'_i|$ decreases; else, the decision makers can increase the threshold C_{sp} when the phenomenon is less critical.

5.3.3/ Temporal Correlation Between Sensors

In addition to the local temporal correlation in each sensor, the readings collected by nearby sensor nodes can be also temporally correlated. The temporal correlation among sensor nodes is to find out the sensors that collect similar readings at the same time in a period. Therefore, it is important that the CH exploits the inter-nodes temporal correlation in order to eliminate the redundancy and improve the network lifetime.

The similarity metrics, such as Euclidean distance and Cosine distance, is one of the methods which can be used to identify sensor nodes that are temporally correlated. These metrics are generally used at the CHs level. Once high temporal correlation between two sensors is found, sensed readings of these sensors are considered redundant. In this case, the CH should schedule these sensors in order to remove the redundancy in the network. In this chapter, we focus on the Euclidean distance which is widely used in various domains.

Let us first consider two data sets R'_i and R'_j generated by the two sensor nodes S_i and S_j respectively in the same period p. Then, in order to compute the Euclidean distance between R'_i and R'_j, we must retransform, similarly to this one in 4.4, the set R'_i (resp. R'_j) to a vector as follows:

$$R'_i = [\ \underbrace{r'_1, \ldots, r'_1}_{wgt(r'_1)\text{ times}}, \underbrace{r'_2, \ldots, r'_2}_{wgt(r'_2)\text{ times}}, \ldots, \underbrace{r'_k, \ldots, r'_k}_{wgt(r'_k)\text{ times}}\ ]$$

where $|R'_i| = |R'_j| = \mathcal{T}$.

Finally, we can calculate the Euclidean distance between the two vectors R'_i and R'_j based on the following equation:

$$E_d(R'_i, R'_j) = \sqrt{\sum_{k=1}^{\mathcal{T}} (r'_{i_k} - r'_{j_k})^2}, \quad \text{where } r'_{i_k} \in R'_i \ \text{ and } r'_{j_k} \in R'_j$$

5.3.3.1/ Distance Normalization

As mentioned in Section 4.4.3, data must be normalized in order to scale all data vectors to have the same variation before comparing them. We recall the Gaussian normalization process as follows: first, we calculate the Euclidean distance for each pair of data vectors in the network:

$$\mathbb{E}_d = \{E_d(R'_1, R'_2), E_d(R'_1, R'_3), \ldots, E_d(R'_{N-1}, R'_N)\}$$

where N is the total number of sensors. Then, we can apply the Gaussian normalization using the following formula:

$$E'_d(R'_i, R'_j) = \frac{E_d(R'_i, R'_j) - \overline{Y}}{6 \times \sigma} + \frac{1}{2} \tag{5.2}$$

where $\overline{Y}$ is the mean of all distances and σ is the standard deviation of pairwise distance over all data.

Thus, R'_i and R'_j are said to be redundant if $E'_d(R'_i, R'_j) \leq C_{tp}$, where C_{tp} is a user defined threshold for the temporal correlation.

5.3.4/ Spatial-Temporal Correlation Between Sensors

Since nearby nodes tend to be correlated in both space and time [150], exploring spatial-temporal correlation of sensed data is an emerging topic in sensor networks that can reduce the energy consumption in data collection. The spatial-temporal correlation in PSNs happens when two nodes that are close geographically take similar readings in a period. In this section, our objective is to search all pairs of sensors that are spatially-temporally correlated then to switch, in a later time, some sensors to the sleep mode in order to reduce redundant sensing and communication.

Based on equations 5.1 and 5.2, we say that two sensors S_i and S_j, collecting the set of readings R'_i and R'_j respectively, are spatially-temporally correlated at the period p if and only if:

$$E_g(S_i, S_j) \times E'_d(R'_i, R'_j) \leq C_{sptp} \tag{5.3}$$

where C_{sptp} is the threshold for the spatio-temporal correlation and it is defined $C_{sptp} = C_{sp} \times C_{tp}$ such that $E_g(S_i, S_j) < C_{sp}$ and $E'_d(R'_i, R'_j) \leq C_{tp}$.

Algorithm 9 describes our technique to find pairs of sensors that are spatially-temporally correlated. The CH searches which neighbors of each sensor S_i are spatially (line 6) and temporally (line 8) correlated with S_i.

Algorithm 9: Spatial-Temporal Correlation Algorithm

Data: Set of sensors: $\mathbb{S} = \{S_1, S_2 \ldots S_N\}$, Set of their readings sets: $\mathbb{R} = \{R'_1, R'_2 \ldots R'_N\}$, C_{sp}, C_{tp}

Result: All pairs of sensors (S_i,S_j) that are spatially-temporally correlated

```
1  $L \leftarrow \emptyset$;
2  for each sensor $S_i \in \mathbb{S}$ do
3      for each sensor $S_j \in \mathbb{S}$ such that $S_j \neq S_i$ do
4          compute $E_g(S_i, S_j)$;
5          if $(E_g(S_i, S_j) \leq 2 \times S_r)$ then
6              if $(E_g(S_i, S_j) \leq C_{sp})$ then
7                  compute $E'_d(R'_i, R'_j)$;
8                  if $(E'_d(R'_i, R'_j) \leq C_{tp})$ then
9                      $L \leftarrow L \cup \{(S_i, S_j)\}$;
10                 end
11             end
12         end
13     end
14 end
```

5.4/ Sleep Scheduling Strategies

After having searched all pairs of spatially-temporally correlated sensors into a cluster, we propose, in this section, two scheduling strategies that allow sensors to work alternatively.

The first strategy is based on the set cover problem while the second one takes into account the correlation degree and the residual energy of sensors when searching the set of active sensors. In each strategy, a set of sensor nodes is selected in each period, based on some criteria, to collect the data in the network while the other sensors will be switched to the sleep mode.

5.4.1/ Set Cover (SC) Strategy

The first strategy for scheduling sensor nodes is based on the Set Cover (SC) problem. In general, the SC problem consists in finding the minimum number of sets that can cover every element in a given universe. Some real-world applications of SC problem include railway and airline crew scheduling, network discovery and phasor measurement unit placement [161]. In our case of PSNs, we apply the SC over the set of correlated sensors in order to divide all sensors into disjoint sensor subsets where every subset is able to completely cover the whole area of interest.

The SC problem can be formally defined in this paper as follows:

Given a set of N sensors $\mathbb{S} = \{S_1, S_2, \ldots, S_N\}$ and the list $\mathbb{L} = \{(S_i, S_j)/E_g(S_i, S_j) \times E'_d(R'_i, R'_j) \leq C_{sptp}\}$ of all pairwise spatially-temporally correlated sensors. Let $\mathcal{A}$ be a binary matrix of $N \times N$ size, whose elements a_{ij} may contain a value from $\{0, 1\}$; 1 means that the sensors (S_i,S_j) are spatially-temporally correlated; 0 otherwise. The a_{ij} is located in i^{th} row and j^{th} column, with $i,j = \{1, \ldots, N\}$. The goal of SC is to find the list $\mathbb{D}$ which contains all the subsets $\mathbb{X} \subseteq \mathbb{S}$, such that each row i is covered by at least one column $j \in \mathbb{X}$. This means that, each subset of sensors $\mathbb{X}$ will cover, in terms of spatial-temporal correlation, all the sensors in $\mathbb{S}$.

More mathematically, the SC can be formulated as a binary integer programming problem as follows:

$$
\begin{aligned}
\textit{Minimize} \quad & \sum_{j=1}^{N} S_j \\
\textit{Subject to} \quad & \sum_{j=1}^{N} a_{ij} S_j \geq 1, \quad i = \{1, \ldots, N\} \\
& S_j \in \{0, 1\}, \quad j = \{1, \ldots, N\}
\end{aligned}
$$

Our set cover (SC) strategy operates in rounds where each round equals $|\mathbb{D}|$ periods, $|\mathbb{D}|$ represents the total number of subsets $\mathbb{X} \subseteq \mathbb{D}$. In each period in the round, the sensors in only one subset will be active while switching the remaining sensors into sleep mode. We active all the subsets simultaneously in the round. After that, a new list $\mathbb{D}$ of subsets $\mathbb{X}$ must be searched, by using SC, for the next round.

Illustrative example: we consider a set of 6 sensors: $\mathbb{S} = \{S_1, S_2, S_3, S_4, S_5, S_6\}$, with the list of spatially-temporally correlated sensors: $\mathbb{L} = \{(S_1, S_2), (S_1, S_3), (S_1, S_4), (S_1, S_5), (S_2, S_6), (S_3, S_4), (S_3, S_6), (S_4, S_5)\}$. This leads to the following mathematically formulation of SC problem:

$$\begin{array}{lccccccccccccc}
\textit{Minimize:} & S_1 & + & S_2 & + & S_3 & + & S_4 & + & S_5 & + & S_6 & & \\
\textit{Subject to:} & S_1 & + & S_2 & + & S_3 & + & S_4 & + & S_5 & & & \geq & 1 \\
 & S_1 & + & S_2 & & & & & & & + & S_6 & \geq & 1 \\
 & S_1 & & & + & S_3 & + & S_4 & & & + & S_6 & \geq & 1 \\
 & S_1 & & & + & S_3 & + & S_4 & + & S_5 & & & \geq & 1 \\
 & S_1 & & & & & + & S_4 & + & S_5 & & & \geq & 1 \\
 & & & S_2 & + & S_3 & & & & & + & S_6 & \geq & 1
\end{array}$$

By applying the SC problem [66], there are at most two feasible solutions where each sensor S_i equals to 1 in at most one solution:

- *Solution 1*: $S_1 = S_6 = 1$ and $S_2 = S_3 = S_4 = S_5 = 0$.
- *Solution 2*: $S_2 = S_4 = 1$ and $S_1 = S_3 = S_5 = S_6 = 0$.

Therefore, we can divide $\mathbb{S}$ into two disjoint subsets of sensors as follows: $\mathbb{L} = \{L_1 = \{S_1, S_6\}, L_2 = \{S_2, S_4\}\}$. Consequently, the current round will consist, by applying our SC strategy, in three periods where in each period the sensors in one subset L_i wil be active. Otherwise, all the sensors are active in the first period. Figure 5.2 shows the active sensors in each period in the round.

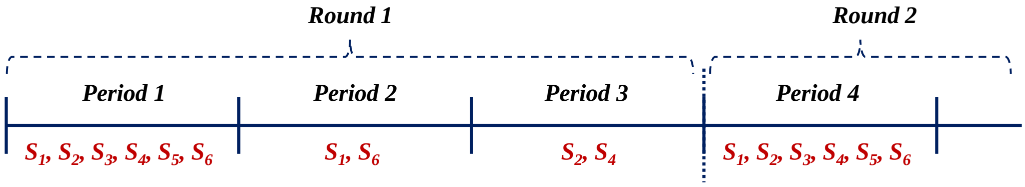


Figure 5.2: Active sensors during periods in the round.

Finally, we notice that the SC strategy is applied by the CH nodes. When receiving the sets of readings from all its sensors in the cluster, in the first period in the round, the CH applies SC strategy to find the active sensors in each period. Then it sends a $sleep$ message to other sensors to switch in sleeping mode. Once the message is received by a sensor, it switches itself to the sleep mode in the next period. On the other hand, we allow the decision makers to determine an upper bound for the number of periods in the rounds if it is necessary and depending on the observed condition.

5.4.2/ Correlation Degree and Residual Energy (CDRE) Strategy

The SC strategy as described did not take into account two important metrics, the residual energy of the sensors and the spatial-temporal correlation degree. For instance, if we fix $C_{sptp} = 5$, two correlated pairs (S_1,S_2) and (S_3,S_4) with the correlation 1 and 4 respectively are treated by the same manner. To overcome this issue, we propose a new strategy for scheduling activity of the sensors based on the degree of the spatial-temporal correlation between sensors and their residual energy. We call this strategy as Correlation Degree and Residual Energy (CDRE) strategy.

The CDRE strategy also operates into rounds where each round is always equal to two periods. In the first period of each round, the CH searches the set of sensors to be active in the second period, based on the CDRE strategy. Given the following notations:

- The list of spatially-temporally correlated sensors with their correlation degrees: $\mathbb{L} = \{((S_i, S_j), C_{ij}(S_i, S_j))\ such\ that\ C_{ij}(S_i, S_j) = E_g(S_i, S_j) \times E'_d(R'_i, R'_j)\ and\ C_{ij}(S_i, S_j) < C_{sptp}\}$.
- The residual energy of a sensor S_i is represented by E_{r_i}.

The CDRE strategy can be expressed using the Algorithm 10. First, we order the pairs of sensors by increasing order of their spatial-temporal correlation degree in order to start with the pair of sensors which have high correlation degree (line 2). Then, in the first period in each round (line 3), we select, in each pair (S_i,S_j), the sensor which has the higher residual energy to be an active sensor in the second period, whereas, the second sensor will be in sleep mode (lines 4-14). The idea behind this selection is to balance the residual energy of the sensors in the network. In the case that a sensor does not have any correlation with other sensors, it must be in active mode always (lines 15-19). After that, we only send the readings sets of the active sensors to the sink (lines 20-22). The objective here is to remove the redundancy among the data sent to the sink in the first period, contrarily to the SC strategy which sends all the readings sets. At the end of the first period, the CH sends a *sleep* message to the sensors which will be switched to the sleep mode in the second period (lines 23-25). Once the message is received by a sensor, it switches itself to the sleep mode in the second period (lines 29-31).

Illustrative example: Recall the sensors S_1 to S_6 in the set $\mathbb{S}$ in the example above with the ordered list of correlated pairs degree as follows: $\mathbb{L} = \{((S_1, S_3), C_{1,3} = 0.5), ((S_4, S_5), C_{4,5} = 0.9), ((S_3, S_4), C_{3,4} = 1.8), ((S_3, S_6), C_{3,6} = 2.1), ((S_1, S_5), C_{1,5} = 2.5), ((S_1, S_2), C_{2,6} = 2.9), ((S_1, S_4), C_{1,4} = 3.3), ((S_2, S_6), C_{1,2} = 3.5)\}$. Then, we consider that the sensors have the following residual energies at the beginning of the round i: $E_{r_1} = 8.1\,mJ$, $E_{r_2} = 8.3\,mJ$, $E_{r_3} = 7.6\,mJ$, $E_{r_4} = 6.9\,mJ$, $E_{r_5} = 7.8\,mJ$, $E_{r_6} = 7.9\,mJ$.

- *Step 1*: We start by the correlated pair (S_1,S_3). Since S_1 has more energy than S_3, S_3 will be switched to the sleep mode in the next period while S_1 will be added to the list of active sensors: $\mathbb{E} = \{S_1\}$. Then, we remove the pairs of sensors that contains S_3, i.e. (S_3,S_4) and (S_3,S_6). The remaining elements in $\mathbb{L} = \{((S_4, S_5), C_{4,5} = 0.9), ((S_1, S_5), C_{1,5} = 2.5), ((S_1, S_2), C_{2,6} = 2.9), ((S_1, S_4), C_{1,4} = 3.3), ((S_2, S_6), C_{1,2} = 3.5)\}$.
- *Step 2*: The first element in $\mathbb{L}$, i.e. (S_4,S_5), is treated similarly to (S_1,S_3): we add S_5 to the list of active sensors and we switch S_4 to the sleep mode then, we remove all elements that contains S_4 from $\mathbb{L}$. Therefore, $\mathbb{E} = \{S_1, S_5\}$ and $\mathbb{L} = \{((S_1, S_5), C_{1,5} = 2.5), ((S_1, S_2), C_{2,6} = 2.9), ((S_2, S_6), C_{1,2} = 3.5)\}$.
- *Step 3*: Since S_1 and S_5 are both in $\mathbb{E}$, we remove the pair (S_1,S_5) from $\mathbb{L}$ because they will be both in active mode. Therefore, $\mathbb{L} = \{((S_1, S_2), C_{2,6} = 2.9), ((S_2, S_6), C_{1,2} = 3.5)\}$.
- *Step 4*: Independent from residual energies of the sensors S_1 and S_2, S_2 should be switched to the sleep mode because S_1 will be considered as active sensor in the next period. Hence, we remove elements from $\mathbb{L}$ that contains S_2: $\mathbb{L} = \{\}$.
- *Step 5*: We add the sensor S_6 to the set $\mathbb{E}$ since it does not have any correlated sensor in $\mathbb{E}$.

Finally, the set of active sensors and the readings sets sent from the CH to the sink, at each period in the round i, are shown in Figure 5.3(a) and Figure 5.3(b) respectively. In

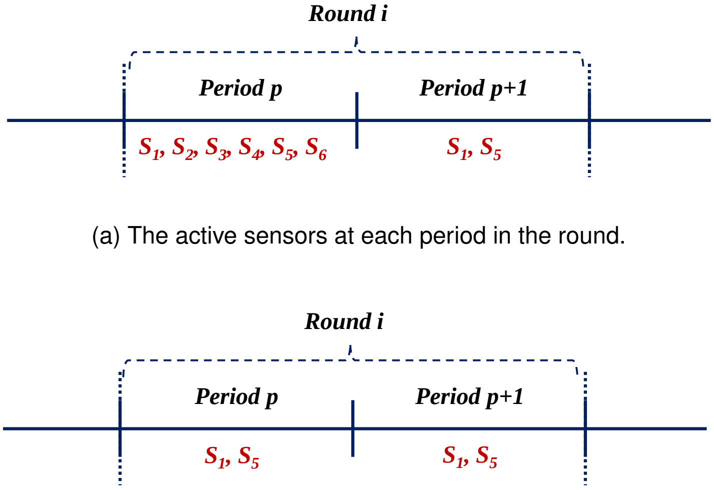
Algorithm 10: CDRE Strategy Algorithm

Data: Set of sensors: $\mathbb{S} = \{S_1, S_2 \ldots S_N\}$, Set of their readings sets: $\mathbb{R} = \{R'_1, R'_2 \ldots R'_N\}$, List of correlated sensors: $\mathbb{L}$, period p.

Result: void

```
$\mathbb{E} \leftarrow \emptyset$;
$\mathbb{L} \leftarrow$ sort($\mathbb{L}$); // $\mathbb{L}$ is sorted in increasing order of the sensors correlation degree;
if $p$ $mod$ 2 is equal to 1 then
    while $\mathbb{L}$ is not empty do
        (($S_i$,$S_j$), $C_{ij}$($S_i$,$S_j$)) is the first element in $\mathbb{L}$;
        consider $E_{r_i} > E_{r_j}$;
        if $S_j$ is not exist in $\mathbb{E}$ then
            if $S_i$ is not exist in $\mathbb{E}$ then
                $\mathbb{E} \leftarrow \mathbb{E} \cup \{S_i\}$;
            end
            remove all elements ($S_y$,$S_j$) and ($S_j$,$S_y$) from $\mathbb{L}$ such that $S_y \neq S_i$;
        end
        remove (($S_i$,$S_j$), $C_{ij}$($S_i$,$S_j$)) from $\mathbb{L}$;
    end
    for each $S_i \in \mathbb{S}$ do
        if $S_i$ has no correlated $S_j$ in $\mathbb{E}$ then
            $\mathbb{E} \leftarrow \mathbb{E} \cup \{S_i\}$;
        end
    end
    for each $S_k \in \mathbb{E}$ do
        $Send\_to\_Sink(R'_k)$;
    end
    for each $S_k \in \mathbb{S}$ such that $S_k \notin \mathbb{E}$ do
        $Sleep\_message\_to(S_k)$;
    end
end
else
    for each $S_k \in \mathbb{S}$ do
        if $Sleep\_message\_to(S_k)$ then
            $S_k$ enter in sleep mode in the current period;
        end
        else
            $Send\_to\_CH(R'_k)$;
        end
    end
end
```

the first period, all the sensors are active while the CH will only send, to the sink, the sets which are not redundant, i.e. corresponding to sensors in the set $\mathbb{E}$. On the other hand, all readings sets coming from the active sensors will be send to the sink in the second period in the round.



(a) The active sensors at each period in the round.

(b) The readings sets sent from the CH to the sink at each period in the round.

Figure 5.3: Illustrative example of the actives sensors and their readings sets during a round.

5.5/ Simulation Results

In this section, we look at the performance of our spatial-temporal correlation mechanism under the two proposed scheduling strategies. In our simulations, we implemented both strategies based on a Java based simulator. We ran the simulator based on real sensor readings of temperature collected by 46 sensors and provided by the Intel Berkeley Research lab [88]. Figure 5.4 shows a map of the placement of sensors in the lab. We assume that the network is divided into two clusters, which have CH$_1$ and CH$_2$ as cluster-heads respectively, as shown in Figure 5.4. The cluster-heads CH$_1$ and CH$_2$ are located at the center of each cluster respectively. The sensor nodes should send their data periodically to their appropriate cluster-head.

In order to evaluate the performance, we compared our results to those of PPMC proposed in [39]. Table 5.1 shows the parameters used in our simulations.

Table 5.1: Simulation environment.

Parameter	Description	Value
$\mathcal{T}$	Number of readings per period	200, 500, 1000
S_r	Sensor sensing range	5, 10, 15, 20 meters
δ	$LocTmp$ similarity threshold	0.03, 0.05, 0.07, 0.1
C_{sp}	Spatial correlation threshold	$2 \times S_r$, $\frac{5 \times S_r}{3}$, $\frac{4 \times S_r}{3}$, S_r
C_{tp}	Temporal correlation threshold	0.35, 0.4, 0.45, 0.5
n	Number of sensors in each cluster	23
t_{PPMC}	Similarity threshold used for PPMC	0.9
E_i	Initial energy for each sensor	10 mJ

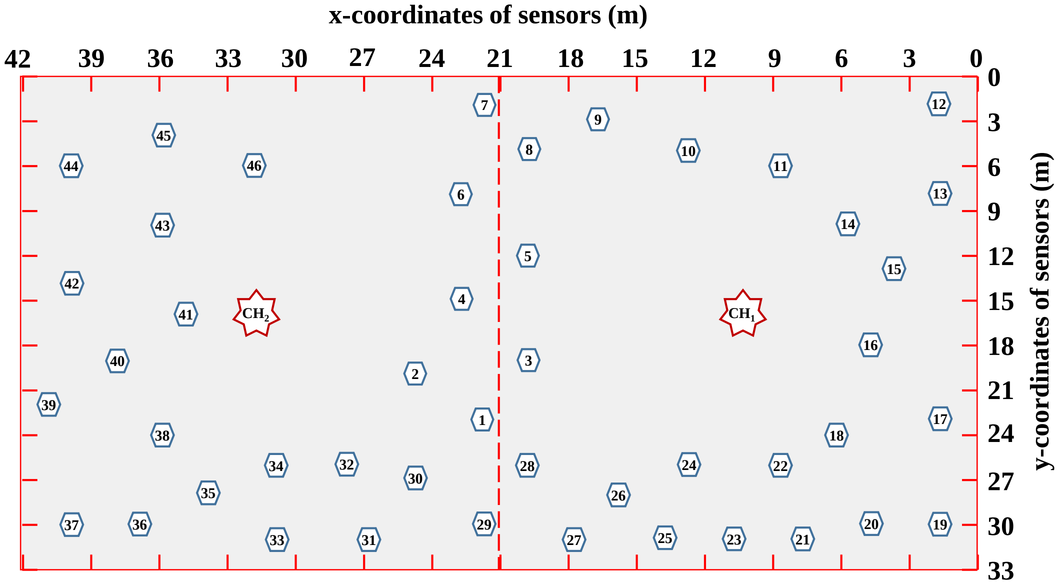


Figure 5.4: Distribution of sensors and CHs in the Intel Laboratory.

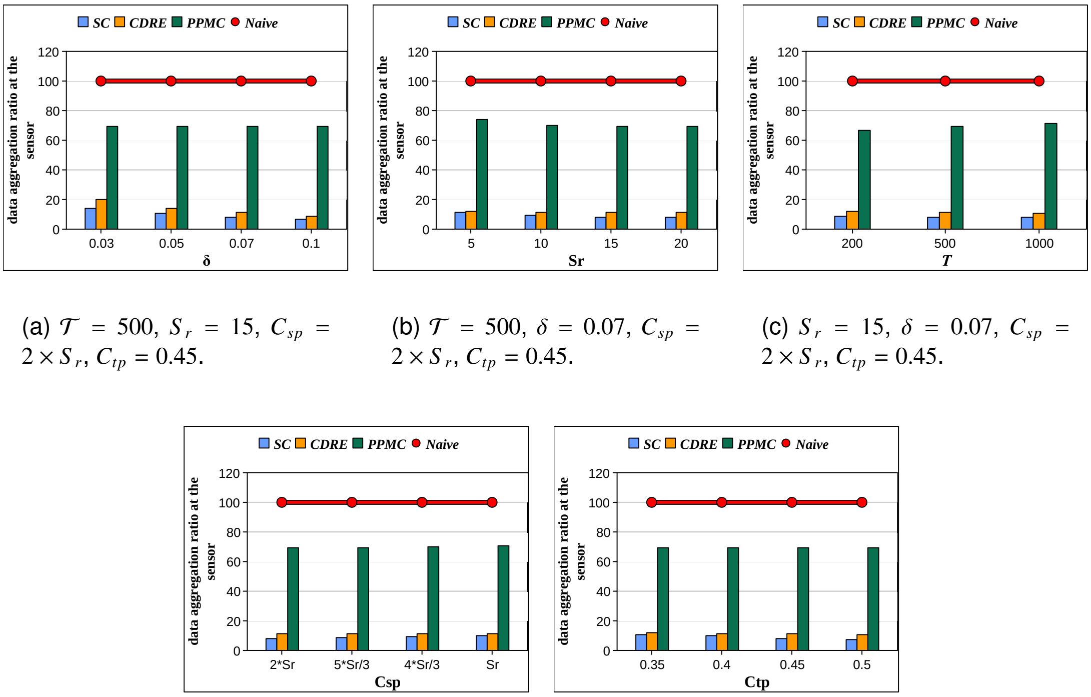
5.5.1/ Performance Evaluation at Sensor Node

In this section, we evaluate the performance of our mechanism with SC and CDRE strategies at the sensor node levels, compared to the PPMC and the naïve method (i.e. the classic method where all readings collected by the sensors are sent to the sink without any processing). We have considered four performance metrics: **(i)** percentage of data readings sent from each sensor to its CH, **(ii)** lifetime of the sensor node, **(iii)** variation of the state and the energy of the sensor during periods, and **(iv)** lifetime of the network in function of active sensors.

Since the two clusters have the same number of sensors which have approximately similar spatial distributions in each of them (see Figure 5.4), similar results for some performance metrics were noticed for the two clusters at the end of the simulation. Hence, we present only the results for one cluster, i.e. the second cluster with CH_2, in the case when the performance metric gives similar results for the two clusters. In addition, the results for each metric shown in the next figures represent the average of all sensors in each cluster.

5.5.1.1/ Percentage of Data Readings Sent from Each Sensor to its CH

In this section, our objective is to show how our mechanism can decrease the data readings collected by each sensor node and then sent to the CH_2. Figure 5.5 shows the percentage of data collected, then sent, by each sensor node when varying one parameter each time and fixing the others as shown in Figure 5.5 (a to e). The obtained results show that PPMC can reduce from 25% to 33% the data sent to the CH_2, while, our mechanism with SC and CDRE strategies can reduce, respectively, up to 93% and 90% the data sent, compared to the naïve technique which always sends all data collected (i.e. 100%) . This means that our approach can effectively eliminate the redundancy in data collection while searching all sensors that generate spatially-temporally correlated data. Furthermore, we can also notice that the SC strategy gives better results, in terms of reducing the data sent by each sensor, than CDRE strategy in all cases. This is because the SC strategy has an objective to search the *minimum* number of sensors in each period that can cover all sensors in the network. Otherwise, CDRE strategy searches an optimal set of sensors in a way that the residual energies of the sensors are balanced in the network while keeping some redundancy level between the selected sensors.



(a) $\mathcal{T} = 500$, $S_r = 15$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

(b) $\mathcal{T} = 500$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

(c) $S_r = 15$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

(d) $\mathcal{T} = 500$, $S_r = 15$, $\delta = 0.07$, $C_{tp} = 0.45$.

(e) $\mathcal{T} = 500$, $S_r = 15$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$.

Figure 5.5: Percentage of data readings sent from each sensor to the CH_2.

Several observations can be made based on the results in Figure 5.5:

- By increasing the threshold δ in Figure 5.5(a), each sensor can reduce, using the SC and CDRE strategies, up to 90% and 87% respectively the readings sent to the CH_2 compared to PPMC. These results are obtained due to the fact that $LocTmp$ will find more similar readings when δ increases.
- By increasing its sensing range as shown in Figure 5.5(b), each sensor sends less readings to the CH_2 using the two proposed strategies. For instance, when S_r increases from 5 to 20, a sensor node decreases its readings sent from 11.4% to 8.4% using SC strategy. This happens because, when S_r increases, each sensor will have more neighboring, thus correlated, sensors. Consequently, more sensors will be switched to the sleep mode, thus, decreasing the percentage of the collected and sent readings.
- By increasing $\mathcal{T}$ from 200 to 1000 in Figure 5.5(c), the percentage of readings sent decreases using SC and CDRE strategies while it increases using PPMC. The rea-

son for this is that the δ threshold used in $LocTmp$ which finds, then eliminates, more redundancy when $\mathcal{T}$ increases in the two strategies. Contrarily, PPMC does not apply any processing on the collected data which increases the readings sent to the CH_2 when $\mathcal{T}$ increases.

- By decreasing the spatial correlation threshold (C_{sp}) in Figure 5.5(d), the percentage of readings sent to the CH_2 increases in the three approaches, i.e. SC, CDRE and PPMC. This result is logical since we make the constraint for the neighboring sensors more difficult to satisfy (see definition 5.3.2). Consequently, the number of active sensors in each period will tend to increase. It is also important to notice that our strategies reduce, in all cases, the readings sent to CH_2 as compared to those sent using PPMC.
- By increasing the temporal correlation threshold (C_{tp}) in Figure 5.5(e), SC and CDRE strategies allow each sensor node to decrease its data readings sent to the CH_2. This is because, the Euclidean distance between sets of readings will be more easily satisfied therefore, more sensors will be switched to the sleep mode.

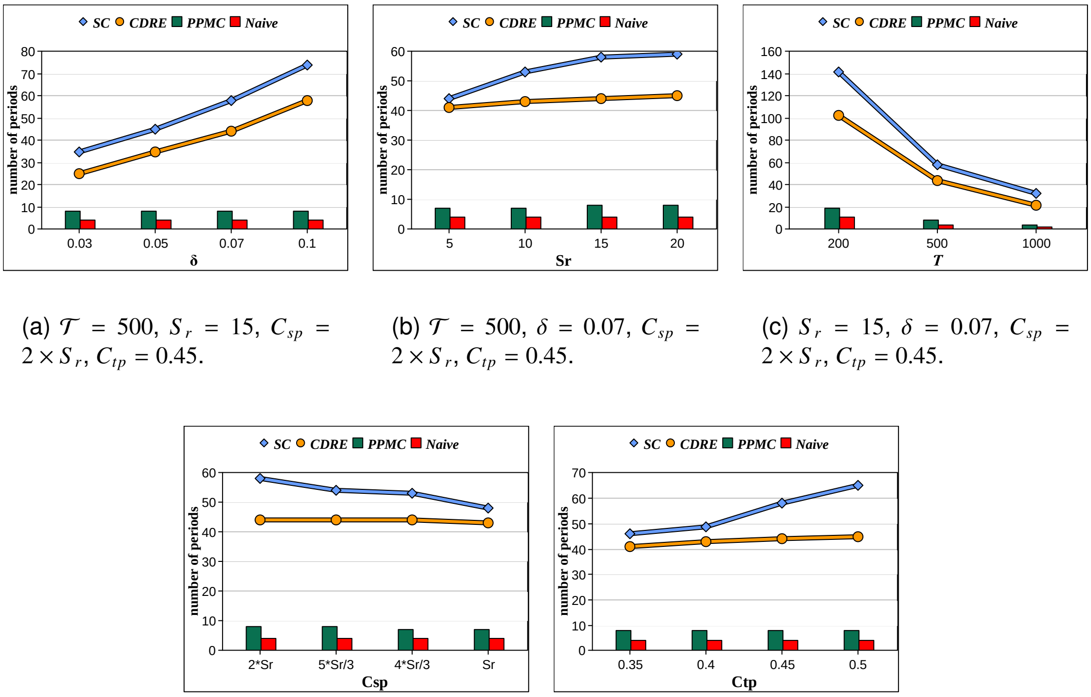
5.5.1.2/ Lifetime of the Sensor Node

In this section, our objective is to study the energy consumption at the sensor nodes level. Therefore, we fixed the initial energy for all sensor nodes to E_i. Then, we applied our strategies, PPMC and Naïve approaches while varying, each time, one parameter and fixing the others as done in Figure 5.5. Figure 5.6 shows the lifetime of each sensor in terms of the number of periods in which the sensor is operational, i.e. its residual energy is positive. The obtained results show clearly that our mechanism, with the proposed strategies, can efficiently reduce the energy consumption of the sensor and extend its lifetime. This is because, our mechanism eliminates the redundancy among collected data and reduces the readings sent to the CH (see Figure 5.5). Although the PPMC can extend, in the best case, the lifetime of a sensor by two times compared to the Naïve approach, our strategies significantly outperform the results of PPMC. We can also notice that, the SC strategy gives better results in terms of keeping the sensor node operating for long time compared to CDRE strategy.

In WSNs, the energy consumption in the sensor node is proportional to the amount of data sent by the sensor. Consequently, when the sensor sends more data the CH, its energy will be more consumed and vice versa. Hence, the observations made based on the results of Figure 5.5 can be similarly made for the energy consumption in the sensor in the Figure 5.6. Table 5.2 shows how many times the sensor node can extend, using our strategies, its lifetime in the worst and the best cases by fixing one parameter as shown in Figure 5.6(a to e), compared to PPMC and Naïve approaches.

5.5.1.3/ Variation of the State and the Energy of the Sensor during Periods

In this section, we show an example of a sensor activitiy variation during periods by applying our strategies, PPMC and Naïve approaches. We take the sensor that has an *id* equals to 35 located in the second cluster, then we study the variation of its state, i.e. active or sleep, and its residual energy during the periods, for some fixed parameters



(a) $\mathcal{T} = 500$, $S_r = 15$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

(b) $\mathcal{T} = 500$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

(c) $S_r = 15$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

(d) $\mathcal{T} = 500$, $S_r = 15$, $\delta = 0.07$, $C_{tp} = 0.45$.

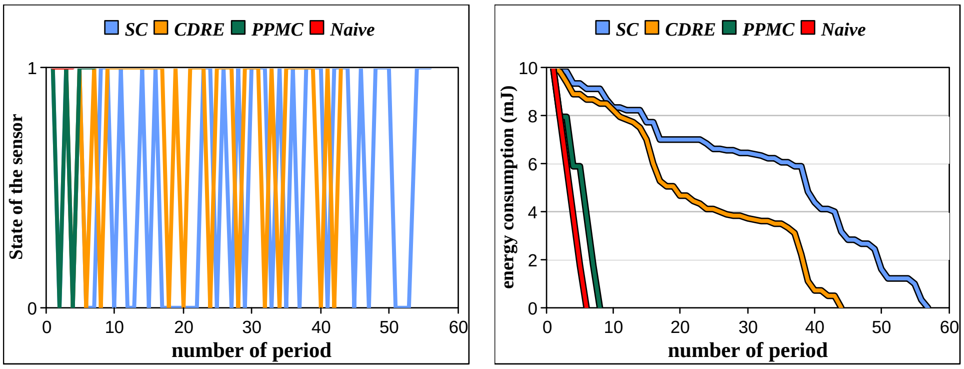
(e) $\mathcal{T} = 500$, $S_r = 15$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$.

Figure 5.6: Lifetime of each sensor in the second cluster (CH_2).

Table 5.2: Lifetime comparisons between our strategies, PPMC and Naïve approaches. (Worst case → Best case).

Our Strategy	*Compared Strategy*	δ	S_r	$\mathcal{T}$	C_{sp}	C_{tp}
SC	*PPMC*	$4 \rightarrow 9$	$6 \rightarrow 7$	$7 \rightarrow 8$	$6 \rightarrow 7$	$5 \rightarrow 8$
	Naïve	$8 \rightarrow 18$	$11 \rightarrow 14$	$12 \rightarrow 16$	$12 \rightarrow 14$	$11 \rightarrow 16$
CDRE	*PPMC*	$3 \rightarrow 7$	$5 \rightarrow 6$	$5 \rightarrow 6$	$5 \rightarrow 6$	$4 \rightarrow 6$
	Naïve	$6 \rightarrow 14$	$10 \rightarrow 11$	$9 \rightarrow 11$	$10 \rightarrow 11$	$10 \rightarrow 11$

shown in Figure 5.7. Based on the results of Figure 5.7(a), we can see that the state of the sensor varies, when applying our strategies, from 1 (i.e. active mode) to 0 (i.e. sleep mode) during the periods more dynamically than with other techniques. Our strategies confirm also the efficient reduction of the redundancy between the sensors correlated to the sensor '35' by switching it to the sleep mode more often than with the other techniques. On the other hand, Figure 5.7(b) shows how the residual energy of the sensor varies depending on the state of the sensor; if the sensor is in active mode, its residual energy decreases in order to collect the data and send it to the CH; else, it remains fixed until the next period. We can also observe that the residual energy of the sensor can remain fixed using the SC strategy in many successive periods, i.e. from periods 18 to 22 in Figure 5.7(b) for example. This happens because the number of periods in each round changes dynamically using the SC strategy where the sensor can be active at most in two periods in a round.



(a) Variation of sensor state.

(b) Variation of sensor energy.

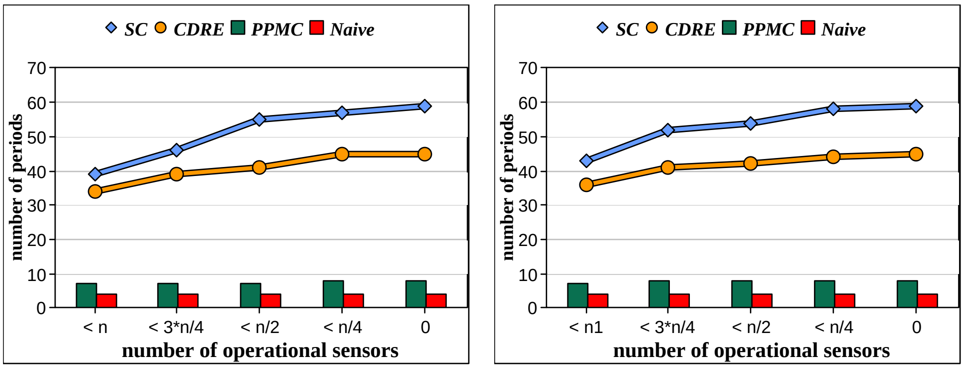
Figure 5.7: Variation of sensor activity during periods, Sensor id = 35, $\mathcal{T} = 500$, $S_r = 15$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

5.5.1.4/ Lifetime of the Network in Function of Active Sensors

Network lifetime is an important metric for the evaluation of sensor networks. Generally, it strongly depends on the number of operational sensor nodes that constitute the network; while this number is greater than a defined threshold, the network is considered efficient (or alive); otherwise, it is considered inefficient. Figure 5.8 shows the lifetime of the two clusters depending on the number of operational sensors in each cluster. We consider that the cluster is always efficient for the following threshold: n, $3 \times n/4$, $n/2$, $n/4$ and 0, where we mean by '$< n$' the first sensor in the network is died and by '0' all the sensors are died. Figures 5.8(a) and 5.8(b) show that the two clusters give similar results for the network lifetime using all strategies. When the threshold for the operational sensors decreases, SC and CDRE strategies extend more the lifetime of each cluster, while the cluster lifetime remains almost fixed using PPMC and Naïve approaches. We can also notice that, when decreasing the threshold from '$< n$' to '0', the cluster lifetime is less extended using CDRE strategy as compared to SC strategy. This happens because CDRE strategy takes into account the residual energies of the sensors to select the set of active sensors. Consequently, this leads to balance the residual energy of sensors during the periods thus, the sensors die simultaneously (in closer periods).

5.5.2/ Performance Evaluation at CH Nodes

In this section, we evaluate the performance of SC and CDRE strategies, PPMC and Naïve approaches at the CH nodes level. We have taken four performance metrics: **(i)** data accuracy, i.e. loss of data readings, **(ii)** variation of the number of periods during rounds, **(iii)** variation of the number of active sensors during periods, **(iv)** illustrative example of data correlation and sensors scheduling, and **(v)** coverage variation during periods.



(a) CH_1

(b) CH_2

Figure 5.8: Lifetime of the network in function of operational sensors, $\mathcal{T} = 500$, $S_r = 15$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

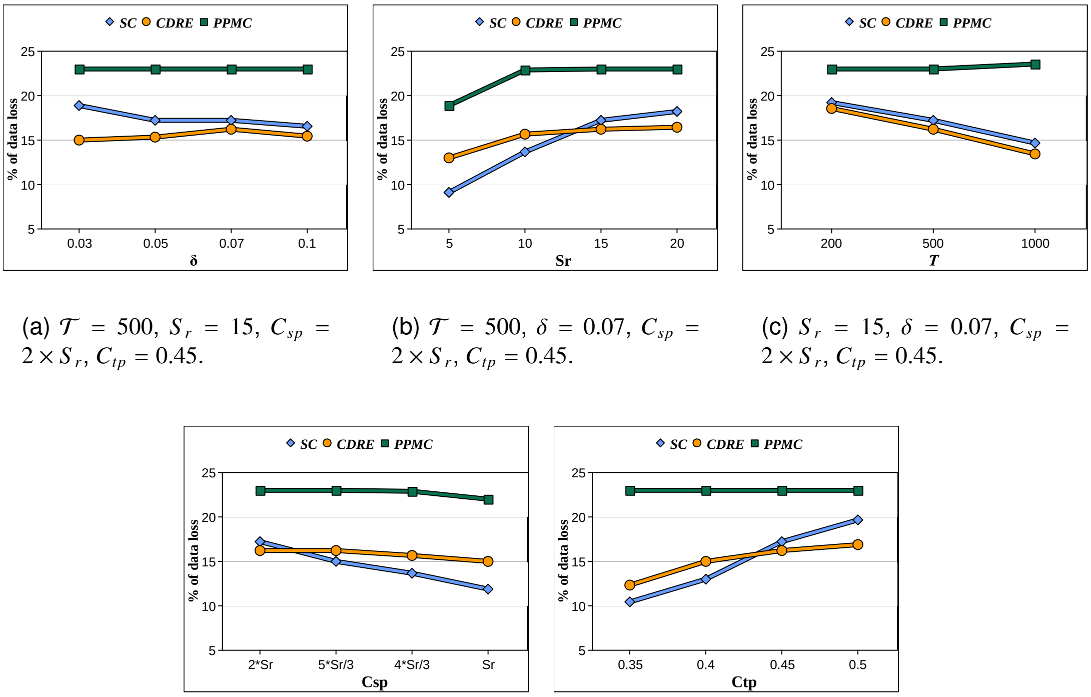
5.5.2.1/ Data Accuracy

Scheduling sensor nodes in the network without losing the integrity of the information is an important challenge for the WSN. Data accuracy represents the loss of readings taken by the sensor nodes whose values (or similar values) do not reach the sink. Figure 5.9 shows the results of data accuracy for SC, CDRE and PPMC for different values of parameters considered in our simulation. We can observe that our strategies always give better results for data accuracy compared to PPMC. This is because, the Pearson coefficient used in PPMC calculates the distance between two data sets based on the summation of readings while the Euclidean distance, used in our strategies, calculates the distance between every two readings in the data sets. This makes the loss of data in our strategies less than that in PPMC. We can also notice that the results of data accuracy using CDRE strategy is better, in most cases, than those obtained using SC strategy. The reason for this is that the sensor sends, using the two strategies, at most two data sets in a round while the round in SC contains more periods than that in CDRE (see illustrative examples for SC and CDRE strategies).

In general, the data accuracy depends on the percentage of data sent by the sensors (see results in Figure 5.5) and on the number of active sensors during the periods; when the data sent to the sink or the number of active sensors increases, the data accuracy increases. Therefore, the following observations can be made based on the results of Figure 5.9: **(1)** data loss increases when the sensing range of the sensor (S_r) or the temporal correlation threshold (C_{tp}) increases (Figures 5.9(b) and 5.9(e)). **(2)** the data accuracy increases when the number of collected readings during a period ($\mathcal{T}$) increases or the spatial correlation threshold (C_{sp}) decreases (Figures 5.9(c) and 5.9(d)).

5.5.2.2/ Variation of the Number of Periods during Rounds

In this section, we show how the number of periods changes, using our proposed strategies, after each round for the two clusters in the network, for some fixed parameters. Using the SC strategy, the CH calculates, at the beginning of each round, the maximum



(a) $\mathcal{T} = 500$, $S_r = 15$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

(b) $\mathcal{T} = 500$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

(c) $S_r = 15$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

(d) $\mathcal{T} = 500$, $S_r = 15$, $\delta = 0.07$, $C_{tp} = 0.45$.

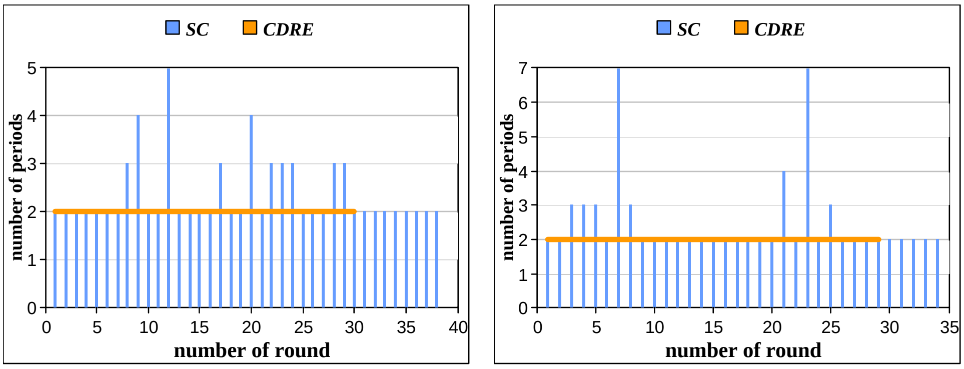
(e) $\mathcal{T} = 500$, $S_r = 15$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$.

Figure 5.9: Data accuracy at the CH_2.

number of periods for the current round based on the set covering problem. Otherwise, the number of periods is always equal to 2 for each round using CDRE strategy. The obtained results of the two clusters, represented by their cluster-heads CH_1 and CH_2 respectively, are shown in Figure 5.10(a) and Figure 5.10(b) respectively. While each round always consists of two periods using CDRE, the number of periods dynamically varies in each round using SC as shown in the figures. We can also observe that: **(1)** the round can contain up to 7 periods using SC strategy. This reflect the high level of redundancy existing in the network where SC can efficiently eliminate this redundancy. **(2)** the sensors in the first cluster are more spatially-temporally correlated compared to those in the second cluster. This leads to extend, using the two strategies, the lifetime of the first cluster more than that of the second cluster.

5.5.2.3/ Variation of the Number of Active Sensors during Periods

In this section, our main goal is to show how our strategies are able to schedule the activities of the sensor nodes for the two clusters. Figure 5.11 shows the number of active sensors in each cluster and in each period using SC and CDRE strategies, for the fixed parameters shown in the figure. The number of active sensors can affect the lifetime of the network, the data latency and the coverage of the monitored area. As we can see, each strategy successfully schedules the sensor nodes in each cluster dynamically after each period according to its own scheduling mechanism. We can notice that, the

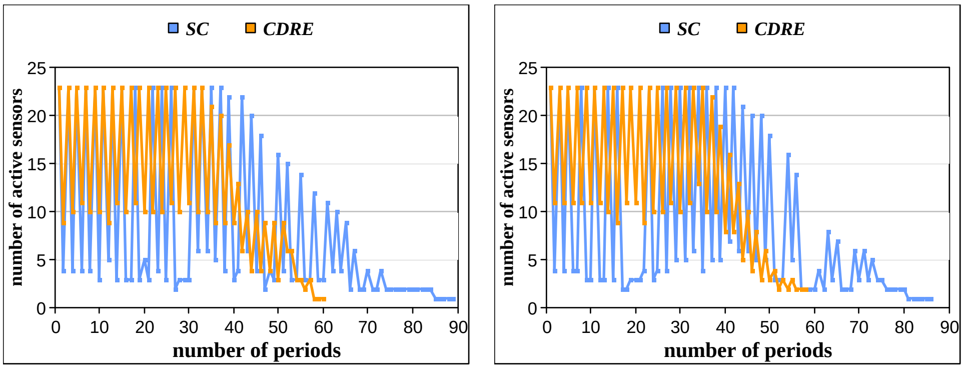


(a) CH_1

(b) CH_2

Figure 5.10: Variation of periods number in each round, $\mathcal{T} = 500$, $S_r = 15$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

SC strategy reduces the number of active sensors, in each cluster, at each period to the minimum while the CDRE strategy selects the set of active sensors that balance the energy distribution in each cluster. Consequently, the obtained results confirm the proper behavior of our strategies.



(a) CH_1

(b) CH_2

Figure 5.11: Variation of active sensors number during each period, $\mathcal{T} = 500$, $S_r = 15$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

5.5.2.4/ Illustrative Example of Data Correlation and Sensors Scheduling

In this section, we show an illustrative example of correlated sensors and how they are scheduled using the two proposed strategies, e.g. SC and CDRE, during a taken period. In this example, we take the sensors in CH_2 (see Figure 5.4) then we fix the parameters as shown in figures 5.12 and 5.13. Based on the figure 5.12, we can see that data generated by the sensors in CH_2 are highly spatio-temporally correlated. Furthermore, we can notice that a sensor is more correlated to its nearest neighboring than the other nodes in the cluster. However, sometimes, correlation between distant nodes can be also seen due to the temporal correlation between their generated data. Finally, we can also observe that some sensor nodes do not have any correlation to other nodes in the cluster, i.e. S_{31}, S_{39}, and S_{45}.

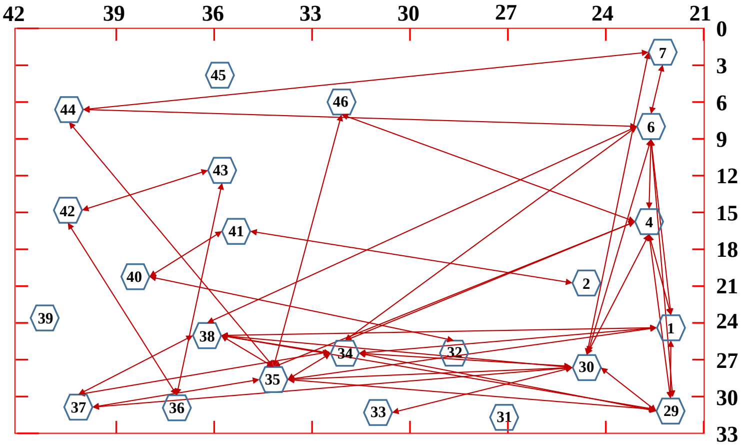
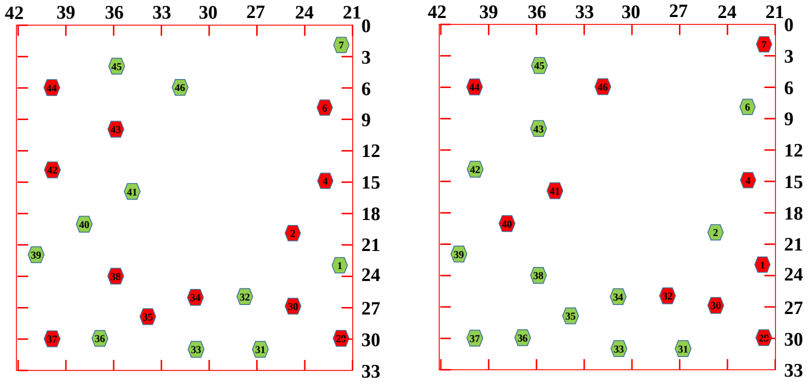


Figure 5.12: E
xample of spatio-temporal data correlation between neighboring nodes during a period, $\mathcal{T} = 500$, $S_r = 10$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.35$.

On the other hand, we show in figure 5.13 how the CH selects the representative nodes for the cluster CH_2 for the next period, using SC and CDRE. We observe that the active sensor nodes are different from one strategy to another.



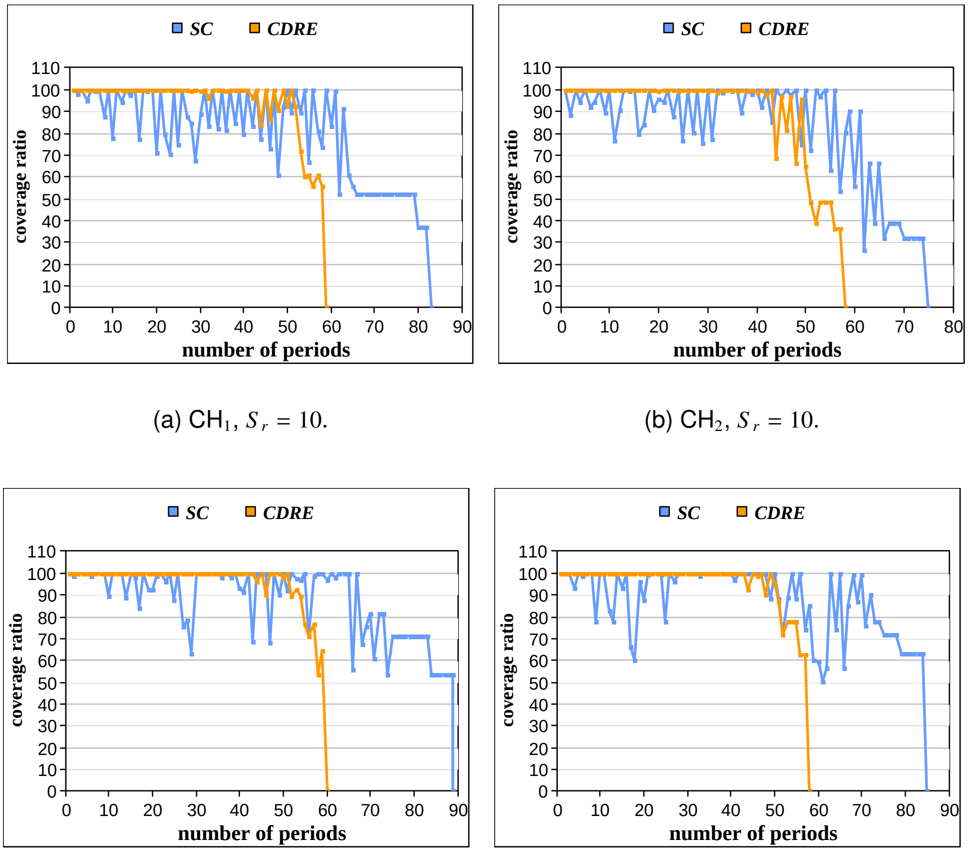
(a) using CDRE

(b) using SC

Figure 5.13: Example of active sensors during a period, $\mathcal{T} = 500$, $S_r = 10$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.35$.

5.5.2.5/ Coverage Variation during Periods

Conserving the network energy while preserving the maximal coverage of the region of interest is an important challenge in WSNs. In Figure 5.14, we show how much of the area of each cluster is covered after each period by applying our strategies. The sensing range of a sensor is varied from 10 in Figures 5.14(a) and 5.14(b) to 15 in Figure 5.14(c) and 5.14(d), while the other parameters remain fixed. The obtained results show that the two proposed strategies provide sufficient coverage for the clusters during each period. Therefore, we can consider that our mechanism with the two proposed strategies can efficiently extend the network lifetime while preserving the integrity of data and the coverage of the observed area.



(a) CH$_1$, $S_r = 10$.

(b) CH$_2$, $S_r = 10$.

(c) CH$_1$, $S_r = 15$.

(d) CH$_2$, $S_r = 15$.

Figure 5.14: Coverage ratio for each cluster, $\mathcal{T} = 500$, $\delta = 0.07$, $C_{sp} = 2 \times S_r$, $C_{tp} = 0.45$.

Based on the results in Figure 5.14, several observations can be made:

- the CDRE strategy provides more coverage for the two clusters compared to SC strategy. This is because, the number of active sensors in each period using CDRE strategy is greater than that using SC strategy.
- the coverage ratio for each cluster increases when the sensor sensing range increases.

5.5.3/ Further Discussions

In this section, we give further consideration to our proposed mechanism. We compare the obtained results for both strategies SC and CDRE. We give some directions to which strategy to choose and under which conditions and circumstances of the application.

From the sensor lifetime point of view, both strategies SC and CDRE significantly improve the lifetime of the sensor (Figure 5.6). However, SC allows sensor to extend more its lifetime, from 7% to 45%, compared to CDRE. Therefore, if the application needs to conserve the energy and extend the network lifetime as long as possible, SC strategy is more suitable.

From the data accuracy point of view, CDRE can save, in most of the cases, the integrity of the collected data more than SC. This is because the number of active sensors in each period using CDRE is greater than that in SC, which increases the accuracy of the data sent to the sink. Consequently, when the priority of the application is to ensure a high level of data accuracy, CDRE is more suitable.

From the coverage of the interest area point of view, CDRE can practically cover the whole monitored area during all periods of the network lifetime, while SC can ensure a satisfactory coverage, i.e. more than 70% in most cases, of the network area. Hence, if the application does not permit flexibility regarding coverage of the network, CDRE is more suitable.

5.6/ Conclusion

In this chapter, we proposed an efficient mechanism in order to search the spatial and temporal correlation between data collected by the sensors in a periodic sensor network. Then, in order to schedule sensors to work alternatively, we proposed two scheduling strategies in order to switch the sensors into sleep/active mode during the periods. The first strategy, called SC, is based on the set cover problem while the second strategy, called CDRE, takes into account the correlation degree and the residual energy of the sensors when scheduling the network. We demonstrated through simulation on real data readings the efficiency of our mechanism, under the two proposed strategies, in sensor networks in terms of extending network lifetime while conserving the quality of the collected data and the coverage of the monitored area.

6

Conclusions and Perspectives

6.1/ Conclusions

Wireless sensor networks are a promising domain for a large variety of applications, such as military and environment monitoring etc. According to MIT Technology Review, WSN is defined one of 10 emerging technologies that will change the world. Indeed, it is not unreasonable to expect that in 10-15 years that the world will be covered with WSNs with access to them via the Internet.

In this thesis, we proposed energy-efficient data management techniques dedicated to periodic sensor networks based on a clustering architecture. We showed that such networks face two major challenges; first, they generate a huge amount of collected data and thus enable complex data analysis for decision makers; second, the energy of sensors will be depleted quickly due to the huge volume of data collection and transmission. Therefore, data management techniques proposed in this thesis were targeted to minimize the amount of data retrieved/communicated by the network without loss in fidelity. The goal of this reduction is first to increase the network lifetime, by optimizing energy consumption of the limited battery for each sensor node, and then to help in analyzing data and making decision. Indeed, we focused on data collection, data aggregation and data correlation in PSNs, with the ultimate goal of extending the network lifetime.

First, we proposed a data collection model that allows each sensor node to adapt its sampling rate to the changing of the monitored condition, in order to minimize the amount of data collected during the collection phase in PSN. Using one-way ANOVA model and statistical tests (Fisher, Tukey and Bartlett), we studied the sensed data inter periods based on the dependence of conditional variance on measurements varies over time. Then, in order to take into account the application criticality, we used an existing multiple level activity model that uses behavior functions modeled by modified Bezier curves to define application classes and allow each node to compute its sampling rate while taking into account its residual energy level. We showed via simulations that our approach can be effectively used to increase the sensor network lifetime, while still keeping the quality of the collected data high.

Second, we proposed a data aggregation and transfer protocol with the main objective is to eliminate redundant data generated in each cluster at both sensors and CH levels. At the first level, we allowed each sensor node to search the similarities between readings collected at each period in order to eliminate redundancy from raw data. Then, it searched duplicated data sets captured among successive periods, using the sets similarity functions, in order to reduce data sets transmission to its CH. At the second level, we proposed a data aggregation technique based on the distance functions in order to allow each CH to find, then eliminate, redundant data sets generated by neighboring nodes, before sending them to the sink. Compared to other existing techniques, we showed through simulations on real data readings the efficiency of our proposed technique in sensor networks in terms of energy consumption, data latency and accuracy.

In a third step, we studied the spatio-temporal data correlation between sensor nodes to exploit the redundancy existing in the network. Based on this correlation, we proposed two sleep/active strategies for scheduling sensors in each cluster. The first one searched the minimum number of active sensors, e.g. which they will collect data, based on the set covering problem while the second one take advantages from the correlation degree and the sensors residual energies for scheduling nodes in the cluster. Our proposed technique, under the two proposed strategies, has been evaluated where the obtained resulted were very encouraged in terms of extending network lifetime, while conserving the quality of the collected data and the coverage of the monitored area.

6.2/ Perspectives

As perspectives of this thesis, we propose two categories. The first one is direct perspectives which are related to the techniques proposed in this work. While the second one is general perspectives and open issues in data management for PSN.

6.2.1/ Direct Perspectives

In this section, we give some perspectives in order to improve, extend or continue the proposed techniques on data collection, data aggregation, data correlation presented in this work.

First, we seek to extend our adaptive sampling technique in order to take into account the correlation between neighboring nodes. Mostly, neighboring sensor nodes collect redundant data about the monitored area thus, the information of “sensor position” should be used when adapting sensor sampling rate. For instance, in the case where sampling rate for two neighboring nodes are adapted in the same manner, “sensor position” information should prevent them to take readings at same slots in the period, or, to put one of them in the sleep mode. On the other hand, collision between packets from two sensor nodes are likely to happen repeatedly when they operate with identical or similar sampling rates. Thus, we seek to adapt our technique to be able to detect this repeated collision and introduce a phase shift between the two transmission sequences in order to avoid further collisions.

Second, we seek to improve the data aggregation process at both sensor and CH levels. At the sensor node level, we plan to use another filtering methods, such as prefix, suffix or position, in order to accelerate the computation of Jaccard similarity condition between two data sets. At the CH level, we seek to optimize the calculation of distance between two data sets in order to minimizing the data latency of the proposed technique. The idea here is to study the minimum number of readings to be used in each sets when calculating the distance between two data sets. Finally, we plan to merge both proposed data aggregation techniques in one work. Indeed, CH uses the distance functions in order to reduce more the number of data sets sent to the sink.

Third, we seek to extend our scheduling technique proposed in Chapter 5 in order to take another information when choosing the number and the set of active sensors in each cluster. An example of information is the number of similar neighboring nodes for a sensor; the sensor that has a high number of similar neighboring nodes must have the priority to switch in sleep mode. Another information is the criticality of the application which is an essential criteria to be used when determining the number of active sensors at each period in order to ensure a high quality for the collected data. Otherwise, we assumed in our technique that the spatial and temporal correlations have the same importance when calculated the spatio-temporal correlation between two nodes which is not always true in all applications. Hence, it is important, as future work, to give a coefficient for each of them depending from the requirements of the application. Finally, the set cover problem used in our technique does not give usually the optimal solution regarding the disjoint sets of active sensors. Consequently, one of the important perspective for this technique is to improve the set cover problem in order to give a near optimal solution, or, to try, in the case it is not possible, another algorithm in integer linear programming (ILP) that gives optimal disjoint sets solution.

Finally, it is interesting to perform real experiments in order to evaluate the performance of our proposed techniques in real world applications.

6.2.2/ General Perspectives and Open Issues

Although the huge number of studies dedicated to data management in periodic sensor networks, the area is still largely open to research. Several key open research issues in data management are yet unexplored or, sometimes, need to be more explored. Researchers' attention should pay more attention to these issues in order to improve the performance of such networks, especially in data analysis, decision making, and energy consumption.

First, wireless and especially periodic sensor networks are collecting huge amounts of data from different fields, and due to their limited memory, computation capabilities, and battery lifetime, it is difficult to sense, store, transfer, and analyze such amount of big data. Nowadays, recent advances in big data are allowing huge amounts of data to be properly captured, structured, processed, and stored. Therefore, big data technology is complementing these smart sensor networks. Thus integrating these two technologies will enable various useful applications. As a future work, we may focus on trying to apply big data methods and analytics to manage periodic sensor data gathered in different areas such as traffic, environment, healthcare, and industrial and designing novel solutions to the challenging problems.

Second, mobility is another issue which remain largely unexplored in sensor networks, especially in periodic applications. Today, the increasing capabilities of mobile tiny sensors and devices make mobile sensor networks possible and practical. By introducing mobility to some or all the nodes in a WSN, we can enhance its capability and enable various applications where mobility plays a key role in its execution. Although wireless sensor networks were never considered to be fully static and many researchers proposed solutions, however mobility still regarded as having several challenges that need to be studied, including, energy consumption, dynamic topology, cluster formation, data aggregation and transmission, among others.

Finally, time synchronization is a significant challenge in PSNs. Since data should be sent periodically, any loss or delayed can change the data time synchronization at the sink which raises a problem in decision making. Therefore, more techniques need to be proposed in order to guarantee an accurate time information for the collected data in PSNs.

Publications

Accepted and Published Journals

[**1**] Hassan Harb, Abdallah Makhoul, and Raphaël Couturier. An Enhanced K-means and ANOVA-based Clustering Approach for Similarity Aggregation in Underwater Wireless Sensor Networks. *IEEE Sensors Journal*, IEEE Publisher, Vol. 15, Iss. 10, pages 5483–5493, 2015.

[**2**] Hassan Harb, Abdallah Makhoul, David Laiymani, Ali Jaber, and Oussama Bazzi. An Analysis of Variance-based Methods for Data Aggregation in Periodic Sensor Networks. *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXII (TLDKS)*, Springer Berlin Heidelberg Publisher, pages 165–183, 2015.

[**3**] Abdallah Makhoul, Hassan Harb, and David Laiymani. Residual Energy-based Adaptive Data Collection Approach for Periodic Sensor Networks. *Journal of Ad Hoc Networks*, Elsevier Publisher, Vol. 35, pages 149–160, 2015.

[**4**] Abdallah Makhoul, David Laiymani, Hassan Harb, and Jacques Bahi. An Adaptive Scheme for Data Collection and Aggregation in Periodic Sensor Networks. *International Journal of Sensor NETworks (IJSNET)*, Inderscience Publishers (IEL), Vol. 18, Iss. (1-2), pages 62–74, 2015.

[**5**] Hassan Harb, Abdallah Makhoul, Ali Jaber, Ramy Tawil, and Oussama Bazzi. Adaptive Data Collection Approach based on Sets Similarity Function for Saving Energy in Periodic Sensor Networks. *International Journal of Information Technology and Management (IJITM)*, Inderscience Publishers (IEL), Vol. *, Iss. *, pages *–*, 2015. Accepted manuscript. To appear.

[**6**] Hassan Harb, Abdallah Makhoul, Ramy Tawil, and Ali Jaber. Energy-Efficient Data Aggregation and Transfer in Periodic Sensor Networks. *IET Wireless Sensor Systems journal*, IET Publisher, Vol. 4, Iss. 4, pages 149–158, 2014.

Conference Articles

[**1**] Hassan Harb, Abdallah Makhoul, Ali Jaber, Samar Tawbi, and Ramy Tawil. Optimized Algorithm for Periodic Data Aggregation in Wireless Sensor Networks. In OCOSS 2013, Ocean & Coastal Observation: Sensors and observing systems, numerical models & information Systems, Nice, France, pages 28–31, October 28-31, 2013.

[**2**] Hassan Harb, Rami Tawil, Ali Jaber and Abdallah Makhoul. Filtering Techniques for Data Aggregation in Periodic Sensor Networks. In LAAS 20, 20th International

Science Conference Advanced Research for Better Tomorrow, Hadath, Lebanon, March 27-29, 2014.

[**3**] Hassan Harb, Abdallah Makhoul, Rami Tawil and Ali Jaber. A Suffix-Based Enhanced Technique for Data Aggregation in Periodic Sensor Networks. In IWCMC 2014, 10th IEEE International Conference on Wireless Communications & Mobile Computing, Nicosia, Cyprus, pages 494–499, August 4-8, 2014.

[**4**] Hassan Harb, Abdallah Makhoul, David Laiymani, Ali Jaber, and Rami Tawil. K-means based clustering approach for data aggregation in periodic sensor networks. In WiMob 2014, 10th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Larnaca, Cyprus, pages 434–441, October 8-10, 2014.

[**5**] Hassan Harb, Abdallah Makhoul, Maguy Medlej, and Raphaël Couturier. An Aggregation and Transmission Protocol for Conserving Energy in Periodic Sensor Networks. In WETICE 2015, 24th IEEE International Conference Enabling Technologies: Infrastructure for Collaborative Enterprises, 5th Track on Cyber Physical Society with SOA, BPM and Sensor Networks, Larnaca, Cyprus, pages 134–139, June 15-17, 2015.

Bibliography

[1] Testing more than two independent groups, http://www.imb.uq.edu.au/download/sfb-lecture-notes-5.pdf.

[2] Anbarasan A., S. Sivasubramaniam, and Mohanasundhram M. A minimum cost effective cluster algorithm using uwsn. *International Journal of Innovative Research in Science, Engineering and Technology*, 3(7):14656–14661, 2014.

[3] Ameer Ahmed Abbasi and Mohamed Younis. A survey on clustering algorithms for wireless sensor networks. *Journal of Computer Communications*, 30((14-15)):2826–2841, 2007.

[4] Haydar Abdulameer Marhoon, M. Mahmuddin, and Shahrudin Awang Nor. Chain-based routing protocols in wireless sensor networks: A survey. *ARPN Journal of Engineering and Applied Sciences*, 10(3):1389–1398, 2015.

[5] Louai Al-Awami and Hossam Hassanein. Energy efficient data survivability for wsns via decentralized erasure codes. In *37th Conference on Local Computer Networks (LCN)*, pages 577–584, 2012.

[6] May Kamil Al-Azzawi, Juan Luo, and Renfa Li. Virtual cluster model in clustered wireless sensor network using cuckoo inspired metaheuristic algorithm. *International Journal of Hybrid Information Technology*, 8(4):133–146, 2015.

[7] Abdo Y. Alfakih, Miguel F. Anjos, Veronica Piccialli, and Henry Wolkowicz. Euclidean distance matrices, semidefinite programming, and sensor network localization. *Portugaliae Mathematica*, 68(1):53–102, 2011.

[8] Cesare Alippi, Giuseppe Anastasi, Cristian Galperti, Francesca Mancini, and Manuel Roveri. Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications. In *IEEE International Workshop on Mobile Ad Hoc and Sensor Systems for Global and Homeland Security (MASS-GHS 2007)*, pages 1–6, 2007.

[9] Ahmed S. Alwakeel, Mohamed F. Abdelkader, Karim G. Seddik, and Atef Ghuniem. Exploiting temporal correlation of sparse signals in wireless sensor networks. In *79th IEEE on Vehicular Technology Conference (VTC Spring)*, pages 1–6, 2014.

[10] Navid Amini, Alireza Vahdatpour, Wenyao Xuand, Mario Gerla, and Majid Sarrafzadeh. Cluster size optimization in sensor networks with decentralized cluster-based protocols. *Computer Communication*, 35(2):207–220, 2012.

[11] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568, 2009.

[12] Arvind Arasu, Venkatesh Ganti, and Raghav Kaushik. Efficient exact set-similarity joins. In *Proc. of the 32nd Int. Conf. on Very Large Data Bases (VLDB 2006)*, pages 918–929, 2006.

[13] Muhammad Ayaz, Azween Abdullah, Ibrahima Faye, and Yasir Batira. An efficient dynamic addressing based routing protocol for underwater wireless sensor networks. *Computer Communications*, 35(4):475–486, 2012.

[14] Li B., McClendon R.W., and Hoogenboom G. Spatial interpolation of weather variables for single locations using artificial neural networks. *Transactions of the ASAE*, 47(2):629–637, 2004.

[15] Nagesh Babu V. and Arudra A. Enhancement of secure and efficient data transmission in cluster based wireless sensor networks. *International Journal of Scientific and Research Publications*, 4(6):1–6, 2014.

[16] Miloud Bagaa, Noureddine Lasla, Abdelraouf Ouadjaout, and Yacine Challal. Sedan: Secure and efficient protocol for data aggregation in wireless sensor networks. In *32nd IEEE Conference on Local Computer Networks (LCN)*, pages 1053–1060, 2007.

[17] Miloud Bagaa, Mohamed Younis, Djamel Djenouri, Abdelouahid Derhab, and Nadjib Badache. Distributed low-latency data aggregation scheduling in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 11(3):49 pages, 2015.

[18] Jacques M. Bahi, Abdallah Makhoul, and Maguy Medlej. Data aggregation for periodic sensor networks using sets similarity functions. In *7th IEEE Int. Wireless Communications and Mobile Computing Conference (IWCMC 2011)*, pages 559–564, 2011.

[19] Jacques M. Bahi, Abdallah Makhoul, and Maguy Medlej. Energy efficient 2-tiers weighted in-sensor data cleaning. In *Proceeding Of 5th International Conference on Sensor Technologies and Applications (SENSORCOMM11)*, pages 197–202, 2011.

[20] Jacques M. Bahi, Abdallah Makhoul, and Maguy Medlej. Energy efficient in-sensor data cleaning for mining frequent itemsets. *Sensors and Transducers journal*, 14(2):64–78, 2012.

[21] Jacques M. Bahi, Abdallah Makhoul, and Maguy Medlej. An optimized in-network aggregation scheme for data collection in periodic sensor networks. In *Proc. Of the 11th Int. Conf. on Ad Hoc Networks and Wireless (ADHOC-NOW 2012)*, pages 153–166, 2012.

[22] Jacques M. Bahi, Abdallah Makhoul, and Maguy Medlej. A two tiers data aggregation scheme for periodic sensor networks. *Ad Hoc & Sensor Wireless Networks*, 21((1-2)):77–100, 2014.

[23] Guillermo Barrenetxea, Francois Ingelrest, Gunnar Schaefer, Martin Vetterli, Olivier Couach, and Marc Parlange. Sensorscope: Out-of-the-box environmental monitoring. In *International Conference on Information Processing in Sensor Networks (IPSN'08)*, pages 332–343, 2008.

[24] Omar Batarfi, Radwa El Shawi, Ayman G. Fayoumi, Reza Nouri, Seyed-Mehdi-Reza Beheshti, Ahmed Barnawi, and Sherif Sakr. Large scale graph processing systems: survey and an experimental evaluation. *Cluster Computing journal*, 18(3):1189–1213, 2015.

[25] David Baum. Cio information matters. big data, big opportunity. http://www.oracle.com/us/c-central/cio-solutions/informationmatters/big-data-big-opportunity/index.html.

[26] Roberto J. Bayardo, Yiming Ma, and Ramakrishnan Srikant. Scaling up all pairs similarity search. In *Proc. Of the 16th international conference on World Wide Web*, pages 131–140, 2007.

[27] Amit S. Bhosale, Sanjay R Khajure, and Manish S. Sharma. Efficient data collection in wireless sensor networks using spatial correlation algorithm. *International Journal on Recent and Innovation Trends in Computing and Communication*, 3(2):418–423, 2015.

[28] N. Boopal, S. Gunasekaran, and V. Alamelu Mangai. A survey of spatiotemporal data compression in wireless sensor networks. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 4(4):1182–1185, 2015.

[29] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29((8-13)):1157–1166, 1997.

[30] H. Brody. 10 emerging technologies that will change the world. *MIT Technology Review*, 106(1):33–49, 2003.

[31] Kang Cai, Gang Wei, and Huifang Li. Information accuracy versus jointly sensing nodes in wireless sensor networks. In *IEEE Asia Pacific conference on circuit and systems*, pages 1050–1053, 2008.

[32] Chih-Min Chao and Tzu-Ying Hsiao. Design of structure-free and energy-balanced data aggregation in wireless sensor networks. *Journal of Network and Computer Applications*, 37:229–239, 2014.

[33] Huifang Chen, Hiroshi Mineno, and Tadanori Mizuno. Adaptive data aggregation scheme in clustered wireless sensor networks. *Computer Communications*, 31(15):3579–3585, 2009.

[34] Siguang Chen, Chuanxin Zhao, Meng Wu, Zhixin Sun, and Jian Jin. Clustered spatio-temporal compression design for wireless sensor networks. In *24th IEEE International Conference on Computer Communication and Networks (ICCCN)*, pages 1–6, 2015.

[35] Yung-Kuei Chiang, Neng-Chung Wang, and Chih-Hung Hsieh. A cycle-based data aggregation scheme for grid-based wireless sensor networks. *Sensors Journal*, 14(5):8447–8464, 2014.

[36] Nathaniel Crary, Bin Tang, and Setu Taase. Data preservation in data-intensive sensor networks with spatial correlation. In *Proceedings of the ACM International Workshop on Mobile Big Data (MobiData 2015) in conjunction with Mobihoc 2015*, pages 7–12, 2015.

[37] Floriano De Rango, Nunzia Palmieri, and Simona Ranieri. Spatial correlation based low energy aware clustering (leach) in a wireless sensor networks. *Journal of Advances in Electrical and Electronic Engineering*, 13(4):350–358, 2015.

[38] Michel Marie Deza and Elena Deza. *Encyclopedia of Distances*. Springer, 2009.

[39] Sunil Dhimal and Kalpana Sharma. Energy conservation in wireless sensor networks by exploiting inter-node data similarity metrics. *International Journal of Energy, Information and Communications*, 6(2):23–32, 2015.

[40] Alexandros G. Dimakis, Vinod Prabhakaran, and Kannan Ramchandran. Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, pages 15–20, 2005.

[41] Tao Du, Zhe Qu, Qingbei Guo, and Shouning Qu. A high efficient and real time data aggregation scheme for wsns. *International Journal of Distributed Sensor Networks*, 2015(2015):11 pages, 2015.

[42] Ozlem Durmaz Incel, Amitabha Ghosh, and Bhaskar Krishnamachari. *Scheduling algorithms for tree-based data collection in wireless sensor networks*, volume Theoretical aspects of distributed computing in sensor networks. Springer Berlin Heidelberg, 2011.

[43] Kai-Wei Fan, Sha Liu, and Prasun Sinha. Dynamic forwarding over tree-on dag for scalable data aggregation in sensor networks. *IEEE Transactions on Mobile Computing*, 7(10):1271–1284, 2008.

[44] Menahem Friedman, Mark Last, Yaniv Makover, and Abraham Kandel. Anomaly detection in web documents using crisp and fuzzy-based cosine clustering methodology. *Information Sciences*, 177:467–475, 2007.

[45] Miguel Garcia, Sandra Sendra, Jaime Lloret, and Raquel Lacuesta. Saving energy with cooperative group-based wireless sensor networks. *In Cooperative Design, Visualization, and Engineering*, 6240:73–76, 2010.

[46] Bugra Gedik, Ling Liu, and Philip S. Yu. Asap: an adaptive sampling approach to data collection in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(12):1766–1783, 2007.

[47] Alia Ghaddar, Tahiry Razafindralambo, Isabelle Simplot-Ryl, David Simplot-Ryl, Samar Tawbi, and Abbas Hijazi. Investigating data similarity and estimation through spatio-temporal correlation to enhance energy efficiency in wsns. *Ad Hoc & Sensor Wireless Networks*, 16(4):273–295, 2012.

[48] Fernando Gielow, Michele Nogueira, and Aldri Santos. Data similarity aware dynamic node clustering in wireless sensor networks. *Ad Hoc Networks*, 24:29–45, 2015.

[49] G. Girban and M. Popa. A glance on wsn lifetime and relevant factors for energy consumption. In *International Joint Conference on Computational Cybernetics and Technical Informatics (ICCC-CONTI)*, pages 523–528, 2010.

[50] Nitin Goyal, Mayank Dave, and Anil Kumar Verma. Fuzzy based clustering and aggregation technique for under water wireless sensor networks. In *In International Conference on Electronics and Communication Systems (ICECS)*, pages 1–5, 2014.

[51] Ismo Hakala, Jukka Ihalainen, Ilkka Kivela, and Merja Tikkakoski. Evaluation of environmental wireless sensor network - case foxhouse. *International Journal on Advances in Networks and Services*, 3((1-2)):29–39, 2010.

[52] Richard Hall. Psychology world, 1998. [online data], http://web.mst.edu/psyworld/tukeysexample.htm.

[53] Nweke F. Henry and Ogbu N. Henry. Wireless sensor networks based pipeline vandalisation and oil spillage monitoring and detection: Main benefits for nigeria oil and gas sectors. *The SIJ Transactions on Computer Science Engineering & its Applications (CSEA)*, 3(1):1–6, 2015.

[54] Monika Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *Proc. Of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 284–291, 2006.

[55] Stacey Higginbotham. Sensor networks top social networks for big data. *https://gigaom.com/2010/09/13/sensor-networks-top-social-networks-for-big-data-2/*, 2010.

[56] Timothy C. Hoad and Justin Zobel. Methods for identifying versioned and plagiarized documents. *Journal of the American Society for Information Science and Technology*, 54(3):203–215, 2003.

[57] Gerrit Hoogenboom. The georgia automated environmental monitoring network. In *Proceedings of the 1993 Georgia Water Resources Conference*, pages 398–402, 1993.

[58] Chun-Chieh Hsiao, Yi-Jing Sung, Seng-Yong Lau, Chia-Hui Chen, Fei-Hsiu Hsiao, Hao-Hua Chu, and Polly Huang. Towards long-term mobility tracking in ntu hospital's elder care center. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 649–654, 2011.

[59] Kim Hye-Young. An energy-efficient load balancing scheme to extend lifetime in wireless sensor networks. *Cluster Computing*, 19(1):279–283, 2016.

[60] Houssein I Assaad, Lan Zhou, Raymond J Carroll, and Guoyao Wu. Rapid publication-ready ms-word tables using one-way anova. *SpringerPlus*, 3(474):1–8, 2014.

[61] Ali Kadhum Idrees, Karine Deschinkel, Michel Salomon, and Raphael Couturier. Coverage and lifetime optimization in heterogeneous energy wireless sensor networks. In *13th Int. Conf. on Networks (ICN)*, pages 49–54, 2014.

[62] Muhammad Imran, Athanasios Vasilakos, Nadeem Javaid, Mohsin Raza Jafri, Zahoor Ali Khan, and Nabil Alrajeh. Chain-based communication in cylindrical underwater wireless sensor networks. *Sensors Journal*, 15:3625–3649, 2015.

[63] Mahammad Irfan Shaik and Sayeed Yasin. Optimal converge cast methods for tree- based wsns. *International Journal of Modern Engineering Research (IJMER)*, 3(4):2585–2587, 2013.

[64] Pekka Iso-Ketola, Tapio Karinsalo, and Jukka Vanhala. Hipguard: A wearable measurement system for patients recovering from a hip operation. In *Second International Conference on Pervasive Computing Technologies for Healthcare (Pervasive-Health)*, pages 196–199, 2008.

[65] Ankur Jain and Edward Y. Chang. Adaptive sampling for sensor networks. In *Proceeedings of the 1st international workshop on Data management for sensor networks (DMSN 2004)*, pages 10–16, 2004.

[66] David S. Jonhson. Approximation algorithms for combinatorial problem. *Journal of Computer and System Sciences*, 9(3):256–278, 1974.

[67] K. Karuppasamy and V. Gunaraj. Optimizing sensing quality with coverage and lifetime in wireless sensor networks. *International Journal of Engineering Research & Technology*, 2(2):1–7, 2013.

[68] Hyung-Sin Kim, Jin-Seok Han, and Yong-Hwan Lee. Scalable network joining mechanism in wireless sensor networks. In *In Proceeding of the IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet'12)*, pages 45–48, 2012.

[69] Linghe Kong, Mingyuan Xia, Xiao-Yang Liu, Min-You Wu, and Xue Liu. Data loss and reconstruction in sensor networks. In *IEEE Proceedings on INFOCOM*, pages 1654–1662, 2013.

[70] Xiaoyan Kui, Shigeng Zhang, Jianxin Wang, and Jiannong Cao. Energy balanced clustering data collection based on dominating set in wireless sensor networks. In *Proceedings of the 2012 IEEE International Conference on Communications (ICC'12)*, pages 193–197, 2012.

[71] Anurag Kumar, P. Vijay Kumar, Bharadwaj Amrutur, G. K. Ananthasuresh, Navakanta Bhat, R. C. Hansdah, Malati Hegde, Joy Kuri, Vinod Sharma, Y.N. Srikant, and Rajesh Sundaresan. Wireless sensor networks for human intruder detection. *Journal of the Indian Institute of Science*, 90(3):347–380, 2010.

[72] Dilip Kumar. Performance analysis of energy efficient clustering protocols for maximising lifetime of wireless sensor networks. *IET Wireless Sensor Systems*, 4(1):9–16, 2014.

[73] Rakesh Kumar and Navdeep Singh. A survey on data aggregation and clustering schemes in underwater sensor networks. *International Journal of Grid Distribution Computing*, 7(6):29–52, 2014.

[74] Surender Kumar, M. Prateek, N.J. Ahuja, and Bharat Bhushan. Meecda: Multihop energy efficient clustering and data aggregation protocol for hwsn. *International Journal of Computer Applications*, 88(9):28–35, 2014.

[75] Jeril Kuriakose, V. Amruth, and N. Swathy Nandhini. A survey on localization of wireless sensor nodes. In *International Conference on Information Communication and Embedded Systems (ICICES)*, pages 1–6, 2014.

[76] David Laiymani and Abdallah Makhoul. Adaptive data collection approach for periodic sensor networks. In *In 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1448–1453, 2013.

[77] Tuan Le Dinh, Wen Hu, Pavan Sikka, Peter Corke, Leslie Overs, and Stephen Brosnan. Design and deployment of a remote robust sensor network: experiences from an outdoor water quality monitoring network. In *In 32nd IEEE Conference on Local Computer Networks (LCN 2007)*, pages 799–806, 2007.

[78] Guorui Li and Ying Wang. Automatic arima modeling-based data aggregation scheme in wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2013(1):1–13, 2013.

[79] Changlei Liu and Guohong Cao. Spatial-temporal coverage optimization in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 10(4):465–478, 2011.

[80] Kezhong Liu, Yang Zhuanga, Zhibo Wang, and Jie Ma. Spatiotemporal correlation based fault-tolerant event detection in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2015(2015):14 pages, 2015.

[81] Zhidan Liu, Wei Xing, Bo Zeng, Yongchao Wang, and Dongming Lu. Distributed spatial correlation-based clustering for approximate data collection in wsns. In *27th International Conference on Advanced Information Networking and Applications (AINA 2013)*, pages 56–63, 2013.

[82] Zhixiong Liu, Jianxin Wang, Shigeng Zhang, Huafu Liu, and Xi Zhang. A cluster-based false data filtering scheme in wireless sensor networks. *Adhoc & Sensor Wireless Networks*, 23((1-2)):21–45, 2014.

[83] Jaime Lloret, Miguel Garcia, Jesus Tomas, and Joel J.P.C. Rodrigues. Architecture and protocol for intercloud communication. *Information Sciences*, 258:434–451, 2014.

[84] Ching-Hu Lu and Li-Chen Fu. Robust location-aware activity recognition using wireless sensor network in an attentive home. *IEEE Transactions on Automation Science and Engineering*, 6(4):598–609, 2009.

[85] Yao Lu, Ioan Sorin Comsa, Pierre Kuonen, and Beat Hirsbrunner. Dynamic data aggregation protocol based on multiple objective tree in wireless sensor networks. In *Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–7, 2015.

[86] Junhai Luo and Jiyang Cai. A dynamic virtual force-based data aggregation algorithm for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2015(2015):7 pages, 2015.

[87] Yajie Ma, Yike Guo, Xiangchuan Tian, and Moustafa Ghanem. Distributed clustering-based aggregation algorithm for spatial correlated sensor networks. *IEEE Sensors Journal*, 11(3):641–648, 2011.

[88] Samuel Madden. Intel berkeley research lab. *http://db.csail.mit.edu/labdata/labdata.html*, 2004.

[89] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, 2002.

[90] Abdallah Makhoul. *Réseaux de capteurs : localisation, couverture et fusion de données*, volume Thesis at the University of Franche-Comté. Besançon, 2008.

[91] R.B. Manjula and Sunilkumar. S. Manvi. Cluster based data aggregation in underwater acoustic sensor networks. In *In Proceeding of the 2012 Annual IEEE India Conference (INDICON'12)*, pages 104–109, 2012.

[92] Kirk Martinez, Royan Ong, and Jane Hart. Glacsweb: a sensor network for hostile environments. In *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pages 81–87, 2004.

[93] Mohammad Masdari and Maryam Tanabi. Multipath routing protocols in wireless sensor networks: A survey and analysis. *International Journal of Future Generation Communication and Networking*, 6(6):181–192, 2013.

[94] Alireza Masoum, Nirvana Meratnia, and Paul J.M. Havinga. An energy-efficient adaptive sampling scheme for wireless sensor networks. In *8th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 231–236, 2013.

[95] Maguy Medlej. *Big data management for periodic wireless sensor networks*, volume Thesis at the University of Franche-Comté. HAL Id : tel-01228515, 2014.

[96] Konstantin Mikhaylov, Jouni Tervonen, Joni Heikkila, and Janne Kansakoski. Wireless sensor networks in industrial environment: Real-life evaluation results. In *2nd Baltic Congress on Future Internet Communications (BCFIC)*, pages 1–7, 2012.

[97] Peyman Mirhadi, Sajjad Zandinia, Azadeh Goodarzipour, Siamak Salimi, and Hossein Goodarzipour. Ip2p k-means: an efficient method for data clustering on sensor networks. *Management Science Letters*, 3(3):967–972, 2013.

[98] Subhas Chandra Mukhopadhyay. Wearable sensors for human activity monitoring: A review. *IEEE Sensors Journal*, 15(3):1321–1330, 2015.

[99] Raymond Mulligan and Habib M. Ammari. Coverage in wireless sensor networks: A survey. *Network Protocols and Algorithms*, 2(2):27–53, 2010.

[100] Hemavathi Natarajan and Sudha Selvaraj. A fuzzy based predictive cluster head selection scheme for wireless sensor networks. In *Proceeding of the 8th International Conference on Sensing Technology*, pages 560–567, 2014.

[101] NCSS. Chapter 210, one-way analysis of variance, http://www.ncss.com/wp-content/themes/ncss/pdf/procedures/ncss/one-way-analysis-of-variance.pdf.

[102] Daniel N. Nkwogu and Alastair R. Allen. Adaptive sampling for wsan control applications using artificial neural networks. *Journal of Sensor and Actuator Networks*, 1(3):299–320, 2012.

[103] Nooshin Nokhanji and Zurina Mohd Hanapi. A survey on cluster-based routing protocols in wireless sensor networks. *Journal of Applied Sciences*, 14(18):2011–2022, 2014.

[104] Rabia Noor Enam, Rehan Qureshi, and Syed Misbahuddin. A uniform clustering mechanism for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2014(2014):14 pages, 2014.

[105] Ali Norouzi, Faezeh Sadat Babamir, and Zeynep Orman. A tree based data aggregation scheme for wireless sensor networks using ga. *Wireless Sensor Network*, 4(8):191–196, 2012.

[106] Luis M.L. Oliveira and Joel J.P.C. Rodrigues. Wireless sensor networks: a survey on environmental monitoring. *Journal of Communications*, 6(2):143–151, 2011.

[107] Abin Abraham Oommen, C.Senthil Singh, and M. Manikandan. Design of face recognition system using principal component analysis. *International Journal Of Research In Engineering And Technology*, 3(1):6–10, 2014.

[108] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T.S. Huang. Supporting similarity queries in mars. In *in Proceedings of the 5th ACM International Multimedia Conference*, pages 403–413, 1997.

[109] K. Ovaliadis and N. Savage. Cluster protocols in underwater sensor networks: a research review. *Journal of Engineering Science and Technology Review*, 7(3):171–175, 2014.

[110] G. Padmavathi and D. Shanmugapriya. A survey of attacks, security mechanisms and challenges in wireless sensor networks. *International Journal of Computer Science and Information Security (IJCSIS)*, 4((1-2)):1–9, 2009.

[111] Ajit R. Pagar and D.C. Mehetre. A survey on energy efficient sleep scheduling in wireless sensor network. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(1):557–562, 2015.

[112] Xinglin Piao, Yongli Hu, Yanfeng Sun, Baocai Yin, and Junbin Gao. Correlated spatio-temporal data collection in wireless sensor networks based on low rank matrix approximation and optimized node sampling. *Sensors Journal*, 14(12):23137–23158, 2014.

[113] Bartlomiej Placzek and Marcin Bernas. Uncertainty-based information extraction in wireless sensor networks for control applications. *Ad Hoc Networks*, 14:106–117, 2014.

[114] GL Prakash, M Thejaswini, SH Manjula, KR Venugopal, and LM Patnaik. Tree-on dag for data aggregation in sensor networks. *Journal Of World Academy of Science, Engineering and Technology*, 37(1):95–101, 2009.

[115] Argo Project. 2000. [online data], http://www.argo.ucsd.edu/index.html.

[116] Lofar project. 2005. http://www.lofar.org/agriculture/fighting-phytophtora-using-micro-climate/fighting-phytophtora-using-micro-climate.

[117] Gang Qian, Shamik Sural, Yuelong Gu, and Sakti Pramanik. Similarity between euclidean and cosine angle distance for nearest neighbor queries. In *Proceeding of the 2004 ACM symposium on Applied computing*, pages 1232–1237, 2004.

[118] Lei Quan, Song Xiao, Xiao Xue, and Cunbo Lu. Neighbor-aided spatial-temporal compressive data gathering in wireless sensor networks. *IEEE Communications Letters*, 20(3):578–581, 2016.

[119] Vijay Raghunathan, Curt Schurgers, Sung Park, and Mani B Srivastava. Energy aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, 2002.

[120] Ramesh Rajagopalan and Pramod K. Varshney. Data-aggregation techniques in sensor networks: A survey. *IEEE Communication Surveys & Tutorials*, 8(4):48–63, 2006.

[121] Ketki Ram Bhakare, R.K. Krishna, and Samiksha Bhakare. An energy-efficient grid based clustering topology for a wireless sensor network. *International Journal of Computer Applications*, 39(14):24–28, 2012.

[122] Prakash Ranganathan and Kendall Nygard. Time synchronization in wireless sensor networks: A survey. *International Journal of UbiComp (IJU)*, 1(2):92–102, 2010.

[123] Rakotomalala Ricco. Comparaison de populations, tests parametriques. *Bartlett test, Version 1.2*, pages 27–29, 2013.

[124] Steel Robert, J Torrie, and D Dickey. *Principles and procedures of statistics: a biometrical approach*, volume 3rd edn. McGraw-Hill Companies, 1997.

[125] Rupali Rohankara, C.P. Kattib, and Sushil Kumarc. Comparison of energy efficient data collection techniques in wireless sensor network. *Procedia Computer Science*, 57(2015):146–151, 2015.

[126] Ahmed Salim and Walid Osamy. Distributed multi chain compressive sensing based routing algorithm for wireless sensor networks. *Wireless Networks Journal*, 21(4):1379–1390, 2015.

[127] K.P. Sampoornam and K. Rameshwaran. An efficient data redundancy reduction for sensed data aggregators in sensor networks. *Journal of Scientific & Industrial Research*, 74:29–33, 2015.

[128] Silvia Santini, Benedikt Ostermaier, and Andrea Vitalett. First experiences using wireless sensor networks for noise pollution monitoring. In *Proceedings of the workshop on Real-world wireless sensor networks (REALWSN'08)*, pages 61–65, 2008.

[129] Sharad Saxena, Shailendra Mishra, and Mayank Singh. Clustering based on node density in heterogeneous under-water sensor network. *International Journal of Information Technology and Computer Science (IJITCS)*, 5(7):49–55, 2013.

[130] Sagiroglu Senef and Sinanc Duygu. Big data: A review. In *In International Conference on Collaboration Technologies and Systems (CTS)*, pages 42–47, 2013.

[131] Vural Serdar and Ekici Eylem. On multihop distances in wireless sensor networks with random node locations. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, 9(4):540–552, 2010.

[132] Rajeev K. Shakya, Yatindra Nath Singh, and Nishchal K. Verma. A novel spatial correlation model for wireless sensor network applications. In *9th International Conference on Wireless and Optical Communications Networks (WOCN)*, pages 1–6, 2012.

[133] M. Shanmukhi and O.B.V. Ramanaiah. Cluster-based comb-needle model for energy-efficient data aggregation in wireless sensor networks. In *Applications and Innovations in Mobile Computing (AIMoC)*, pages 42–47, 2015.

[134] Maiying Shen and Shuo Chen. Unequal distributed spatial correlation-based tree clustering for approximate data collection. In *International Conference on Soft Computing in Information Communication Technology (SCICT 2014)*, pages 93–97, 2014.

[135] Yonghui Shim and Younghan Kim. Data aggregation with multiple sinks in information-centric wireless sensor network. In *International Conference on Information Networking (ICOIN 2014)*, pages 13–17, 2014.

[136] Robert Szewczyk, Alan Mainwaring, Joseph Polastre, John Anderson, and David Culler. An analysis of a large scale habitat monitoring application. In *In Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 214–226, 2004.

[137] Deepali Tambekar R. Energy efficient wireless sensor network for big data management. *International Engineering Research Journal (IERJ)*, 1(3):92–95, 2015.

[138] S. Taruna, Rekha Kumawat, and G.N. Purohit. Multi-hop clustering protocol using gateway nodes in wireless sensor network. *International Journal of Wireless & Mobile Networks (IJWMN)*, 4(4):169–180, 2012.

[139] Khoa Thi-Minh Tran and Seung-Hyun Oh. A data aggregation based efficient clustering scheme in underwater wireless sensor networks. *Ubiquitous Information Technologies and Applications, Lecture Notes in Electrical Engineering*, 280:541–548, 2014.

[140] Khoa Thi-Minh Tran, Seung-Hyun Oh, and Jeong-Yong Byun. Well-suited similarity functions for data aggregation in cluster-based underwater wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2013(Article ID 645243):7 pages, 2013.

[141] A. Tiwari, F.L. Lewis, and S.S. Ge. Wireless sensor network for machine condition based maintenance. *In Control, Automation, Robotics and Vision Conference (ICARCV 2004)*, 1:461–467, 2004.

[142] Gilman Tolle, Joseph Polastre, Robert Szewczyk, David Culler, Neil Turner, Kevin Tu, Stephen Burgess, Todd Dawson, Phil Buonadonna, David Gay, and Wei Hong. A macroscope in the redwoods. In *Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys'05)*, pages 51–63, 2005.

[143] Ankit Tripathi, Sanjeev Gupta, and Bharti Chourasiya. Survey on data aggregation techniques for wireless sensor networks. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(7):7366–7371, 2014.

[144] Ming-Hui Tsai and Yueh-Min Huang. A sub-clustering algorithm based on spatial data correlation for energy conservation in wireless sensor networks. *Journal of Sensors*, 14(11):21858–21871, 2014.

[145] Edward N. Udo and Etebong B. Isong. Flood monitoring and detection system using wireless sensor network. *Asian Journal of Computer and Information Systems*, 1(4):108–113, 2013.

[146] Akihide Utani, Shingo Nakagawa, and Hisao Yamamoto. A novel data gathering scheme for monitoring-oriented wireless sensor networks. *International Journal of Innovative Computing, Information and Control*, 9(1):111–122, 2013.

[147] Maryam Vahabi, MFA Rasid, RSAR Abdullah, and MHF Ghazvini. Adaptive data collection algorithm for wireless sensor networks. *International Journal of Computer Science and Network Security (IJCSNS)*, 8(6):125–132, 2008.

[148] Iuliu Vasilescu, Keith Kotay, Daniela Rus, Matthew Dunbabin, and Peter Corke. Data collection, storage and retrieval with an underwater sensor network. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 154–165, 2005.

[149] Leandro A. Villas, Azzedine Boukerche, Horacio A.B.F. de Oliveira, Regina B. de Araujo, and Antonio A.F. Loureiro. A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks. *Ad Hoc Networks*, 12:69–85, 2014.

[150] Leandro A. Villas, Azzedine Boukerche, Daniel L. Guidoni, Horacio A.B.F. de Oliveira, Regina Borges de Araujo, and Antonio A.F. Loureiro. An energy-aware spatio-temporal correlation mechanism to perform efficient data collection in wireless sensor networks. *Journal of Computer Communications*, 36(9):1054–1066, 2013.

[151] Mehmet C. Vuran, Ozgur B. Akan, and Ian F. Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Journal of Computer Networks*, 45(3):245–259, 2004.

[152] Mehmet C. Vuran and Ian F. Akyildiz. Spatial correlation-based collaborative medium access control in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 14(2):316–329, 2006.

[153] Chaonan Wang, Liudong Xing, Vinod M. Vokkarane, and Yan Sun. Reliability of wireless sensor networks with tree topology. *International Journal of Performability Engineering*, 8(2):213–216, 2012.

[154] Fei Wang, Liming Wang, Yan Han, Bin Liu, Jian Wang, and Xinyan Su. A study on the clustering technology of underwater isomorphic sensor networks based on energy balance. *Journal of Sensors*, 14(7):12523–12532, 2014.

[155] Tianming Wang. Research on data aggregation technology based on wireless sensor networks. *International Journal of Future Generation Communication and Networking*, 9(1):127–134, 2016.

[156] Tim Wark, Wen Hu, Peter Corke, Jonathan Hodge, Aila Keto, Ben Mackey, Glenn Foley, Pavan Sikka, and Michael Brunig. Bspringbrook: Challenges in developing a long-term, rainforest wireless sensor network. In *Proceeding of International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 599–604, 2008.

[157] Geoffrey Werner-Allen, Jeff Johnson, Mario Ruiz, Jonathan Lees, and Matt Welsh. Monitoring volcanic eruptions with a wireless sensor network. In *Proceedings of the Second European Workshop on Wireless Sensor Networks*, pages 108–120, 2005.

[158] Rebecca Willett, Aline Martin, and Robert Nowak. Backcasting: adaptive sampling for sensor networks. In *Third International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 124–133, 2004.

[159] Alex L. Wood, Geoff V. Merrett, Steve R. Gunn, Bashir M. Al-Hashimi, Nigel R. Shadbolt, and Wendy Hall. Adaptive sampling in context-aware systems: a machine learning approach. In *IET Conference on Wireless Sensor Systems (WSS)*, pages 1–5, 2012.

[160] Wei Xing, Wei Jie, Dimitrios Tsoumakos, and Moustafa Ghanem. A network approach for managing and processing big cancer data in clouds. *Journal of Cluster Computing*, 18(3):1285–1294, 2015.

[161] Jin Xu, Miles HF Wen, Victor OK Li, and Ka-Cheong Leung. Optimal pmu placement for wide-area monitoring using chemical reaction optimization. In *In Proc. IEEE Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–6, 2013.

[162] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin. A wireless sensor network for structural monitoring. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys'04)*, pages 13–24, 2004.

[163] Zheng Xu, Yunhuai Liu, Neil Yen, Lin Mei, Xiangfeng Luo, Xiao Wei, and Chuanping Hu. Crowdsourcing based description of urban emergency events using social media big data. *IEEE Transactions on Cloud Computing*, PP(99):1, 2016.

[164] Zheng Xu, Yunhuai Liua, Lin Mei, Chuanping Hu, and Lan Chen. Semantic based representing and organizing surveillance big data using video structural description technology. *Journal of Systems and Software*, 102:217–225, 2015.

[165] Zheng Xu, Lin Mei, Yunhuai Liu, Hui Zhang, and Chuanping Hu. Crowd sensing based semantic annotation of surveillance videos. *International Journal of Distributed Sensor Networks*, 2015(Article ID 679314):9 pages, 2015.

[166] Arda Yalcuk and S. Postalcioglu. Evaluation of pool water quality of trout farms by fuzzy logic: monitoring of pool water quality for trout farms. *International Journal of Environmental Science and Technology*, 12(5):1503–1514, 2015.

[167] Guangsong Yang, Mingbo Xiao, En Cheng, and Jing Zhang. A cluster-head selection scheme for underwater acoustic sensor networks. In *Proceeding of International Conference on Communications and Mobile Computing (CMC'10)*, pages 188–191, 2010.

[168] Jun Ye. Cosine similarity measures for intuitionistic fuzzy sets and their applications. *Mathematical and Computer Modelling*, 53((1–2)):91–97, 2011.

[169] Pan Yi, Lizhi Xiao, and Yuanzhong Zhang. Remote real-time monitoring system for oil and gas well based on wireless sensor networks. In *International Conference on Mechanic Automation and Control Engineering (MACE)*, pages 2427–2429, 2010.

[170] Fei Yuan, Yiju Zhan, and Yonghua Wang. Data density correlation degree clustering method for data aggregation in wsn. *IEEE Sensors Journal*, 14(4):1089–1098, 2014.

[171] Liang Zhao and Qilian Liang. Optimum cluster size for underwater acoustic sensor networks. In *Proceeding of the 2006 IEEE conference on Military communications (MILCOM'06)*, pages 1–5, 2006.

[172] Yanfei Zheng, Kefei Chen, and Weidong Qiu. Building representative based data aggregation tree in wireless sensor networks. *Mathematical Problems in Engineering*, 2010(2010):11 pages, 2010.

[173] Jing Zhou, D. De Roure, and S. Vivekanandan. Adaptive sampling and routing in a floodplain monitoring sensor network. In *Proceedings of the 2006 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB)*, pages 85–93, 2006.

[174] Chuan Zhua, Chunlin Zhenga, Lei Shuc, and Guangjie Hana. A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Network and Computer Applications*, 35(2):619–632, 2012.

[175] Pinghui Zou and Yun Liu. A data-aggregation scheme for wsn based on optimal weight allocation. *Journal Of Networks*, 9(1):100–107, 2014.

Document generated with LATEX and:
the LATEX style for PhD Thesis created by S. Galland — http://www.multiagent.fr/ThesisStyle
the tex-upmethodology package suite — http://www.arakhne.org/tex-upmethodology/

Abstract:

In this thesis, we propose energy-efficient data management techniques dedicated to periodic sensor networks based on clustering architecture. First, we propose to adapt sensor sampling rate to the changing dynamics of the monitored condition using one-way ANOVA model and statistical tests (Fisher, Tukey and Bartlett), while taking into account the residual energy of sensor. The second objective is to eliminate redundant data generated in each cluster. At the sensor level, each sensor searches the similarity between readings collected at each period and among successive periods, based on the sets similarity functions. At the CH level, we use distance functions to allow CH to eliminate redundant data sets generated by neighboring nodes. Finally, we propose two sleep/active strategies for scheduling sensors in each cluster, after searching the spatio-temporal correlation between sensor nodes. The first strategy uses the set covering problem while the second one takes advantages from the correlation degree and the sensors residual energies for scheduling nodes in the cluster. To evaluate the performance of the proposed techniques, simulations on real sensor data have been conducted. We have analyzed their performances according to energy consumption, data latency and accuracy, and area coverage, and we show how our techniques can significantly improve the performance of sensor networks.

Keywords: Periodic Sensor Networks, Clustering Architecture, Adaptive Sensor Sampling Rate, Similarity and Distance Functions, Spatio-Temporal Correlation, Scheduling Strategies.

Résumé :

Dans cette thèse, nous proposons des techniques de gestion de données pour économiser l’énergie dans les réseaux de capteurs périodiques basés sur l’architecture de clustering. Premièrement, nous proposons d’adapter le taux d’échantillonnage du capteur à la dynamique de la condition surveillée en utilisant le modèle de one-way ANOVA et des tests statistiques (Fisher, Tukey et Bartlett), tout en prenant en compte l’énergie résiduelle du capteur. Le deuxième objectif est d’éliminer les données redondantes générées dans chaque cluster. Au niveau du capteur, chaque capteur cherche la similarité entre les données collectées à chaque période et entre des périodes successives, en utilisant des fonctions de similarité. Au niveau du CH, nous utilisons des fonctions de distance pour permettre CH d’éliminer les ensembles de données redondantes générées par les nœuds voisins. Enfin, nous proposons deux stratégies actif/inactif pour ordonnancer les capteurs dans chaque cluster, après avoir cherché la corrélation spatio-temporelle entre les capteurs. La première stratégie est basée sur le problème de couverture des ensembles tandis que la seconde prend avantages du degré de corrélation et les énergies résiduelles de capteurs pour ordonnancer les nœuds dans chaque cluster. Pour évaluer la performance des techniques proposées, des simulations sur des données de capteurs réelles ont été menées. La performance a été analysée selon la consommation d’énergie, la latence et l’exactitude des données, et la couverture, tout en montrant comment nos techniques peuvent améliorer considérablement les performances des réseaux de capteurs.

Mots-clés : Réseaux de Capteurs Périodiques, Architecture Clustering, Adaptation de Taux d’échantillonnage de Capteurs, Fonctions de Similarité et de Distance, Corrélation spatio-temporelle, Stratégies d’Ordonnancement.

■ École doctorale SPIM 16 route de Gray F - 25030 Besançon cedex
■ tél. +33 (0)3 81 66 66 02 ■ **ed-spim@univ-fcomte.fr** ■ www.ed-spim.univ-fcomte.fr

UFC
UNIVERSITÉ
DE FRANCHE-COMTÉ